A characterization of graph properties testable for general planar graphs with one-sided error (It's all about forbidden subgraphs)

Artur Czumaj*
DIMAP and Department of Computer Science
University of Warwick
Email: A.Czumaj@warwick.ac.uk

Christian Sohler†
Department of Computer Science
TU Dortmund
Email: christian.sohler@tu-dortmund.de

***Abstract*—The problem of characterizing testable graph properties (properties that can be tested with a number of queries independent of the input size) is a fundamental problem in the area of property testing. While there has been some extensive prior research characterizing testable graph properties in the dense graphs model and we have good understanding of the bounded degree graphs model, no similar characterization has been known for *general graphs*, with *no degree bounds*. In this paper we take on this major challenge and consider the problem of characterizing all testable graph properties in *general planar graphs*.**

We consider the model in which a general planar graph can be accessed by the random neighbor oracle that allows access to any given vertex and access to a random neighbor of a given vertex. We show that, informally, a graph property P is testable with one-sided error for general planar graphs *if and only if* testing P can be reduced to testing for a finite family of finite forbidden subgraphs. While our presentation focuses on planar graphs, our approach extends easily to general minor-free graphs.

Our analysis of the necessary condition relies on a recent construction of canonical testers in the random neighbor oracle model that is applied here to the one-sided error model for testing in planar graphs. The sufficient condition in the characterization reduces the problem to the task of testing H-freeness in planar graphs, and is *the main and most challenging technical contribution of the paper: we show that for planar graphs (with arbitrary degrees), the property of being H-free is testable with one-sided error for every finite graph H, in the random neighbor oracle model*.

***Index Terms*—property testing; H-freeness; general planar graphs; minor-free graphs; constant-time algorithms;**

I. Introduction

The fundamental problem in the area of graph property testing is for a given undirected graph G to distinguish if G satisfies some graph property $\mathcal{P}$ or if G is ε-far from satisfying $\mathcal{P}$, where G is said to be ε-far from satisfying $\mathcal{P}$ if an ε-fraction of its representation should be modified in order to make G satisfy $\mathcal{P}$. The notion of testability of combinatorial structures and of graphs, has been introduced by Goldreich et al. [17], who have shown that many natural graph properties such as k-colorability or having a large clique are *testable*, that is, have a tester, whose query complexity, that is, the number of oracle queries to the input representation (in [17], to the graph adjacency matrix) can be upper bounded by a function that depends only on the property $\mathcal{P}$ and on ε, the proximity parameter of the test, and is independent of the size of the input graph G. This has been later extended to show that testability in the *dense graph model* (of [17]) is closely related to the graph regularity lemma as one can show that a property is testable (with two-sided error) if and only if it can be reduced to testing for a finite number of regular partitions [2]; for one-sided error testing, it has been shown that a property is testable if and only if it is hereditary or close to hereditary [6]. In particular, we know that subgraph freeness is testable with one-sided error in this model (see, e.g., [5]). We also know of similar logical characterization of families of testable graph properties (for example, every first-order graph property of type $\exists\forall$ is testable, while there are first-order graph properties of type $\forall\exists$ that are not testable [1]).

While for many years the main efforts in property testing have been concentrated on the dense graph model, there has been also an increasing amount of research focusing on the *bounded degree graph model* introduced by Goldreich and Ron [18], the model more suitable for sparse graphs. For example, while it is trivial to test the subgraph freeness with one-sided error in this model, testing H-minor freeness is more complex, and is possible with constant query complexity only if H is cycle-free [10]; if H has a cycle, then $\widetilde{\Omega}_\varepsilon(\sqrt{n})$ queries are required and effectively sufficient [10], [15]. Among further highlights, it is known that every hyperfinite property is testable with two-sided error [25] (see also [8], [12], [19]).

Rather surprisingly, much less is known for *general graphs*, that is, graphs with no bound for the maximum degree (see, e.g., [16, Chapter 10]). The model has been initially studied by Kaufman et al. [22], Parnas and Ron [26], and Alon et al. [3], where the main goal was to study the trade-off between the complexity for sparse graphs with that for dense graphs (it should be noted though that these papers were using a slightly different access oracle to the input graph). These results show that most of even very basic properties are *not testable*.

*Research partially supported by the Centre for Discrete Mathematics and its Applications (DIMAP), by IBM Faculty Award, and by EPSRC award EP/N011163/1.
†Research supported by ERC grant No. 307696.

Czumaj et al. [11] addressed a related question in this model, and show that in fact if one restricts the input graphs to be planar (but without any constraints on the maximum degree), then the benchmark problem of testing bipartiteness is testable in the random neighbor query model. In a similar vein, Ito [21] extended the framework from [25] and show that all graph properties are testable for a certain special class of multigraphs called hierarchical-scale-free multigraphs. Still, despite these few results and despite its natural importance, *our understanding of graph property testing for degree-unconstrained graphs is very limited*. In this paper we take on this major challenge and consider the problem of characterizing all testable graph properties in *general planar graphs*. We consider the model in which a general planar graph can be accessed by the *random neighbor oracle* that allows access to any given vertex and access to a random neighbor of a given vertex. We show that, informally, a graph property $\mathcal{P}$ is testable with one-sided error for general planar graphs *if and only if* testing $\mathcal{P}$ can be reduced to testing for a finite family of finite forbidden subgraphs. While our presentation focuses on planar graphs, our approach extends easily to general minor-free graphs.

The central combinatorial problem considered in this paper is that of subgraph detection. The question of identifying frequent subgraphs in big graphs is one of the most fundamental problems in network analysis, extensively studied in the literature. It has been empirically shown that different classes of networks have the same frequent subgraphs and they differ for different network classes [24]. In this context, frequently occurring subgraphs are also known as *network motifs* [24]. This raises the question how quickly we can identify the motifs of a given network. Recent work approaches this question by approximating the number of occurrences of certain subgraphs using random sampling [14], [15], [20]. In this paper, we will study the corresponding property testing question: *Can we distinguish a graph that has no copies of a predetermined subgraph H from a graph in which we need to remove more than an ε-fraction of its edges in order to obtain a graph that contains no copy of H*. This question has received a lot of attention in the property testing setting and it is known that subgraph freeness can be tested with a constant number of queries both in the dense graph model (see, e.g., [5]) and in the bounded degree graph model, where testing subgraph freeness is simple. The problem of testing subgraph freeness has also been studied in the setting of general graphs [3], where the authors give a lower bound of $\Omega(n^{1/3})$ queries for testing triangle freeness, which can be extended to other non-bipartite subgraphs. They also give an upper bound of $O(n^{6/7})$ queries. We continue this line of research, but will put our focus on sparse graphs, i.e., graphs with bounded average degree. Since it seems that for many properties we cannot hope for extremely efficient, that is, testing algorithms with a constant number of queries in general graphs (often a hard example is a clique on $\sqrt{n}$ vertices), we focus our attention on *planar graphs*. It has been only recently shown that bipartiteness in planar graphs can be tested with a constant number of queries [11]. Our result can be viewed as a major extension of that result: we prove that for every fixed graph H, the property of H-freeness can be tested with a constant number of queries. Our approach extends to general *minor-free graphs*.

A. Basic notation

Before we proceed with detailed description of our results, let us begin with some basic definitions.

Throughout the paper we use several constants depending on H (forbidden subgraph) and ε. We use lower case Greek letters to denote constants that are typically smaller than 1 (e.g., $\delta_i(\varepsilon, H)$) and lower case Latin letters to denote constants that are usually larger than 1 (e.g., $f_i(\varepsilon, H)$). All these constants are always positive. Furthermore, throughout the paper we use the asymptotic symbols $O_{\varepsilon,H}(\cdot)$, $\Omega_{\varepsilon,H}(\cdot)$, and $\Theta_{\varepsilon,H}(\cdot)$, which ignore multiplicative factors that depend only on H and ε and that are positive for $\varepsilon > 0$.

Throughout the paper, for any set of edge-disjoint subgraphs $\mathcal{S}$ of $G = (V, E)$, we write $G[\mathcal{S}]$ to denote the graph with vertex set V and edge set being the set of edges from the sets in $\mathcal{S}$.

Property testing and H-freeness: A *graph property* $\mathcal{P}$ is any family of graphs closed under isomorphism. (For example, bipartiteness is a graph property $\mathcal{P}$ defined by a family of all bipartite graphs.) We are interested in finding an algorithm (called *tester*) for testing a given graph property $\mathcal{P}$, i.e., an algorithm that inspects only a very small part of the input graph G, and accepts if G satisfies $\mathcal{P}$ with probability at least $\frac{2}{3}$, and rejects G if it is ε-far away from $\mathcal{P}$ with probability at least $\frac{2}{3}$, where ε is a proximity parameter, $0 \le \varepsilon \le 1$. We say a simple graph G is *ε-far from $\mathcal{P}$* if one has to delete or insert more than $\varepsilon|V|$ edges from G to obtain a graph satisfying $\mathcal{P}$[1].

The main focus of this paper is on the study of testers with *one-sided error*, that is, testers that always accept all graphs satisfying $\mathcal{P}$ and can err only for graphs ε-far from $\mathcal{P}$. (In contrast, *two-sided error* testers can err (with probability at most $\frac{1}{3}$) both for graphs ε-far from $\mathcal{P}$ *and* for graphs satisfying $\mathcal{P}$.)

The main graph properties considered in this paper are related to *forbidden subgraphs*. Throughout the entire paper we will fix $H = (V(E), E(H))$ to be an arbitrary, simple, finite undirected graph. The notion of a graph H being *finite* means that its size is constant, though we will allow the constant to be a function of ε, the proximity parameter for

[1]Similarly as in [11], we notice that the standard definition of being ε-far (see, e.g., the definition in [16] or [22]) expresses the distance as the fraction of edges that must be modified in G to obtain a graph satisfying $\mathcal{P}$; comparing this to our definition, instead of modifying $\varepsilon|V|$ edges, one modifies $\varepsilon|E|$ edges. In this paper we prefer to use the definition with $\varepsilon|V|$ edge modifications because our focus is on the study of sparse graphs, graphs with $|E| = O(|V|)$. Indeed, for any class of planar graphs or graphs with an excluded minor, which are the main classes of graphs studied in this paper, the number of edges in the graph is upper bounded by $O(|V|)$. Moreover, unless the graph is very sparse (i.e., most of its vertices are isolated, in which case even finding a single edge in the graph may take a large amount of time), the number of edges in the graph is $\Omega(|V|)$. Thus, under the standard assumption that $|E| = \Omega(|V|)$, the ε in our definition and the ε in the previous definitions remain within a constant factor. We use our definition of being ε-far for simplicity; our analysis can be extended to the standard definition in a straightforward way.

property testing, which will be clear from the context. (That is, for a given graph property $\mathcal{P}$ and a proximity parameter ε, $0 < \varepsilon < 1$, we will say that a graph H is *finite* (for $\mathcal{P}$ and ε) if there is $s = s(\varepsilon) = O_\varepsilon(1)$, such that $|V(H)| \le s$ for every $n \in \mathbb{N}$.)

We say that a given graph G is *H-free* if G does not contain a copy of H. Following the definition above, we say that a simple graph G is *ε-far from H-free* if one has to delete more than $\varepsilon|V|$ edges from G to obtain an H-free graph.

Our definitions extends to families of forbidden graphs in a natural way. If $\mathcal{H}$ is a finite family of finite graphs, then a given graph G is *$\mathcal{H}$-free* if for every $H \in \mathcal{H}$, G is H-free. Similarly, G is *ε-far from $\mathcal{H}$-free* if for every $H \in \mathcal{H}$, G is ε-far from H-free. Further, notice that if $\mathcal{H}$ is a finite family of finite graphs then since each $H \in \mathcal{H}$ is of size $O_\varepsilon(1)$, so is the size of $\mathcal{H}$; hence, $\mathcal{H}$ is also a *finite family* of *finite graphs*.

In our paper we will also consider the following generalization of $\mathcal{H}$-freeness. In what follows, for a given graph property $\mathcal{P}$ and $n \in \mathbb{N}$, let $\mathcal{P}_n$ be the graph property $\mathcal{P}$ for n-vertex graphs.

Definition 1: **(Semi-subgraph-freeness)** A graph property $\mathcal{P} = (\mathcal{P}_n)_{n\in\mathbb{N}}$ is *semi-subgraph-free* if for every ε, $0 < \varepsilon < 1$, and every $n \in \mathbb{N}$, there is a finite family $\mathcal{H}$ of finite graphs such that the following hold:

(i) any graph G satisfying $\mathcal{P}_n$ is $\mathcal{H}$-free, and
(ii) any graph G which is ε-far from satisfying $\mathcal{P}_n$, is not $\mathcal{H}$-free (contains a copy of some $H \in \mathcal{H}$).

Let us emphasize that in Definition 1 by a *finite* family $\mathcal{H}$ of *finite graphs* we mean that even though $\mathcal{H}$ may depend on n, the sizes of $\mathcal{H}$ and of any $H \in \mathcal{H}$ are always upper bounded by a function independent on n, $|\mathcal{H}| = O_\varepsilon(1)$ and $|V(H)| = O_\varepsilon(1)$.

B. Oracle access model: random neighbor queries

The access to the input graph is given by an *oracle*. We consider the *random neighbor oracle*, in which an algorithm is given $n \in \mathbb{N}$ and access to an input graph $G = (V, E)$ by a query oracle, where $V = \{1, \dots, n\}$. The *random neighbor query* specifies a vertex $v \in V$ and the oracle returns a vertex that is chosen i.u.r. (independently and uniformly at random) from the set of all neighbors of v. (Notice that in the random-neighbor model, since $V = \{1, \dots, n\}$, the algorithm can also trivially select a vertex from V i.u.r.)

We believe that the random-neighbor model is the most natural model of computations in the property testing framework in the context of very fast algorithms, and therefore our main focus is on that model.

Remark 2: We notice that all our results could be also presented in a variant of the model above in which we would allow only two types of queries: *random vertex query*, which returns a random vertex, and *random neighbor query*, which returns a random neighbor of a given vertex v.

Each time we call the random neighbor oracle, the returned random vertex or its random neighbor is chosen *independently and uniformly at random (i.u.r.)*. All vertices of the input graph are accessible and distinguishable by their IDs, and there is no requirement about the IDs other than that they are all distinct. Notice that in this model, the tester does not know n, the size of the input graph G.

Query complexity: The *query complexity* of a tester is the number of oracle queries it makes.

We say a graph property $\mathcal{P}$ is *testable* if it has a *tester with constant query complexity*, that is, for every ε, $0 < \varepsilon < 1$, there is $q = q(\varepsilon)$ such that for every $n \in \mathbb{N}$ the tester has query complexity upper bounded by q (the complexity may depend on $\mathcal{P}$ and ε, but not on the input graph nor its size).

Other oracle access models: There are some natural variations of the random neighbor oracle model that have been considered in the literature and that can be relevant here.

I. One could extend the random neighbor oracle model to the *random distinct neighbor oracle model*, where one allows for every vertex to query for *distinct* random neighbors (that is, each time we call the random distinct neighbor query for a given vertex v, the oracle returns a neighbor of v chosen i.u.r. among all neighbors not returned earlier); if all neighbors have been already returned then the oracle would return a special symbol.
II. One could consider a model allowing two other types of queries: *degree queries:* for every vertex $v \in V$, one can query the degree of v, and *neighbor queries:* for every vertex $v \in V$, one can query its i^{th} neighbor. Observe that by first querying the degree of a vertex, we can always ensure that the i^{th} neighbor of the vertex exists in the second type of query.

It should be noted that while our main focus is on the random neighbor oracle model, our testers (and their analysis) for H-freeness can be trivially modified to work in the other three oracle access models (in particular, Theorems 14, 38, and 40 hold in all these models). However, our main result, the characterization of testable properties in planar graphs cannot be extended to the other models (see Section I-C4), other than the variant of the random neighbor oracle discussed in Remark 2.

C. Our results

In this paper we present a characterization of all testable graph properties for general planar graphs in the random neighbor oracle model, showing that, informally, a graph property $\mathcal{P}$ is testable with one-sided error for general planar graphs if and only if testing $\mathcal{P}$ can be reduced to testing for a finite family of finite forbidden subgraphs (see Theorem 5). Further, the results extend to general families of *minor-free graphs G*.

The result is proven in two steps: First we apply a recent result from [9] (see Theorem 11) to argue in Theorem 12 the (easier) necessary condition, that in the random neighbor oracle model, any graph property $\mathcal{P}$ testable with one-sided error can be reduced to testing for a finite family of forbidden subgraphs. Then we prove *our main technical contribution*, Theorem 14, that for a given connected finite graph H, *subgraph freeness* is *testable* (can be tested with a constant number of queries) on any input *planar graph G*, assuming the access to G is via the random neighbor oracle. This latter result extends to arbitrary (not necessarily connected) finite graphs H and to testing for $\mathcal{H}$-freeness for any finite family $\mathcal{H}$ of finite graphs, see Theorem 38 in Section VIII. By combining these results, our characterization in Theorem 5 of graph properties testable with one-sided error for general planar graphs will follow; this result extends to general minor-free graphs.

While we believe that our general characterization of all testable graph properties of planar and minor-free graphs is a central problem in property testing and is the main contribution of this paper, we also hope that our constant query time tester for subgraph freeness will further advance our understanding of efficient algorithms for that fundamental problem.

Our work is a continuation of our efforts to understand the complexity of testing basic graph properties in graphs with no bounds for the degrees. Indeed, while major efforts in the property testing community have been put to study dense graphs and bounded degree graphs (cf. [16, Chapter 8-9]), we have seen only limited advances in the study of arbitrary graphs, in particular, sparse graphs but without any bounds for the maximum degrees. We believe that this model is one of the most natural models, and it is also most relevant to computer science applications (see also the motivation in [16, Chapter 10.5.3]). While the understanding of testing in general graphs is still elusive, our work makes a major step forward towards understanding of testing properties for most extensively studied classes of graphs, in our case of planar and minor-free graphs.

1) Overview: Any testable property can be reduced to testing for forbidden subgraphs: We begin with an *easier part* of our characterization (see Section II for details). Our approach follows the method of canonical testers for graph properties testable for general graphs developed recently in [9]. The *intuition* here is rather simple: if a graph property $\mathcal{P}$ is testable then all what the tester can do is for a given input graph G to randomly sample a constant number of vertices and then to explore their neighborhoods of constant size, and on the basis of the visited subgraph U of G to decide whether to accept the input graph or to reject it. Further, the assumption that we consider a one-sided error tester implies that the tester must always accept any graph G satisfying $\mathcal{P}$. Therefore, in particular, if we define $\mathcal{H}$ as the family of all U for which the tester rejects any input graph G that contains U, then we can argue that any graph G satisfying $\mathcal{P}$ must be $\mathcal{H}$-free. The analysis can be easily extended to hold for an arbitrary class of the input graphs, e.g., for planar graphs.

(Notice that these arguments show only that any testable graph property $\mathcal{P}$ has a finite family $\mathcal{H}$ of finite graphs such that $\mathcal{P}$ is $\mathcal{H}$-free. However, we do not say anything about any other properties of $\mathcal{P}$; indeed, $\mathcal{P}$ may be not only $\mathcal{H}$-free but also may have some other properties. A good example showing the sensitivity of this notion is testing bipartiteness. It has been shown [11] that for general planar graphs bipartiteness is testable with one-sided error, but clearly, bipartiteness cannot be defined as a property of $\mathcal{H}$-freeness for a *finite* family $\mathcal{H}$ of forbidden graphs. However, one can easily show (cf. [10, Section 2.1]) that if an input graph G is ε-far from bipartiteness, then there must be an odd $k = O(1/\varepsilon^2)$, so that G is $O(\varepsilon)$-far from C_k-free, and this fact suffices to argue that bipartiteness for planar graphs is testable.)

To turn this intuition into a formal proof, we need to do some additional work. We rely heavily on the canonical tester developed recently in [9] to argue that to test any testable graph property we can assume that the tester at hand is "oblivious" and works non-adaptively. This allows us to obtain a clean characterization of forbidden subgraphs for any given testable property $\mathcal{P}$. Further, we lift this characterization to extend the analysis to *semi-subgraph-free* graph properties, which are graph properties defined as $\mathcal{H}$-free or close to $\mathcal{H}$-free, for some finite family $\mathcal{H}$ of finite graphs. The analysis is presented in Section II (see Theorem 12).

2) Overview: Testing for forbidden subgraphs in planar graphs and minor-free graphs: The *main technical contribution* of this paper is a proof that *for planar graphs*, the property of being H-free is testable with one-sided error for every connected finite subgraph H, in the random neighbor oracle model, see Theorem 14. This result extends to arbitrary (not necessarily connected) finite graphs H and to testing for $\mathcal{H}$-freeness for any finite family $\mathcal{H}$ of finite graphs, see Theorem 38. Further, the results extend to general families of *minor-free graphs* G, see Theorem 40.

Let us first discuss the challenges of the task of testing H-freeness. It has been known for a long time that for *bounded degree graphs* one can test H-freeness with a constant number of queries using the following simple tester: randomly sample a constant number of vertices and check whether any of them belongs to a copy of H. This result relies on two properties of bounded degree graphs: *(i)* that it is easy to test whether a given vertex belongs to a copy of H (just run a BFS of depth $|V(H)|$), and *(ii)* that if a given graph is ε-far from H-free then it has many edge-disjoint copies of H that cover a total of a linear number of vertices. But both these properties fail to work for general graphs. For example, for *(ii)*, consider an n-vertex graph G with $n - \sqrt{n}$ isolated vertices and $\sqrt{n}$ vertices forming a clique. It is easy to see that G is ε-far from H-free (for a sufficiently small ε with respect to the size of H), but all copies of H in G are covered only by $\sqrt{n}$ vertices and as the result, testing H-freeness trivially requires $\Omega(\sqrt{n})$ queries: one has to perform so many queries (in expectation) to hit a first non-isolated vertex.

In our analysis, by focusing on planar (or minor-free) graphs, we are able to circumvent the latter obstacle *(ii)* (argued implicitly in Lemma 20), but the former obstacle *(i)* still persists. Our approach to cope with *(i)* is by devising a simple modification of BFS search, random bounded-breadth bounded-depth search. By bounding the breadth and depth of the graph exploration we are able to ensure that the complexity of the tester is bounded. However, then the main challenge in our analysis is to analyze this process, to show that indeed, it distinguishes between H-free graphs and graphs that are ε-far from H-free.

Our approach relies on a proof that for any planar graph G

that is ε-far from H-free there exists a set $\mathbb{Q}$ of edge-disjoint copies of H such that,

(i) if we can find a copy of H in $G[\mathbb{Q}]$ with a constant number of queries, then also in G we can find a copy of H with a constant number of queries, and

(ii) if the input graph was $G[\mathbb{Q}]$, then we could find a copy of H with a constant number of queries.

The construction of the set $\mathbb{Q}$ is existential, and is performed by a process of gradually deleting edges of G so that after each round of edge deletions, *(i)* is maintained, and so that at the end, the structure of $G[\mathbb{Q}]$ is simple enough so that *(ii)* is easy. The process is controlled by a sequence of *contractions*: we reduce the problem of finding a copy of H in G to the problem of finding a copy of H with one vertex contracted, which in turn, we reduce to the problem of finding a copy of H with two vertices contracted, and so on so forth. The idea is that if at the end of this process, we have to find a copy of H contracted to single vertex, this task is easy to analyze. The main challenge of our analysis here is to carefully manage the contractions to have the analysis going through. In a similar context, the authors in [11] have been arguing that this task is already very complex for cycles in the analysis of constant-length random walks in planar graphs, that is, graphs with good separators and bad expansion. However, by using a sequence of self-reductions relying on contractions (and hence reducing testing C_k-freeness to testing C_{k-1}-freeness, where C_k is a cycle of length k), the authors in [11] were able to show there that for planar graphs, *testing bipartiteness* (implicitly, testing C_k-freeness for constant k) can be done with constant query complexity and with one-sided error.

The approach presented in our paper can be seen as a major extension of the approach used for testing bipartiteness in [11] to test H-freeness, though the problem of testing H-freeness is significantly more complex. Indeed, the central tool used for bipartiteness, contractions of a path or a cycle, becomes problematic when the forbidden graph H has vertices of degree higher than 2. The challenge here is that to contract vertices of higher degrees, the information about their neighbors is difficult to be maintained. Still, we follow a similar approach, but since we cannot perform the contraction in term of graphs, we do it via introducing *hyperedges*, to ensure that after contracting high degree vertices the information about their neighbors is memorized in a form of a *hypergraph*. This extension of the framework from graphs to hypergraphs makes the entire analysis significantly more complicated and one of our main technical contributions is to make the analysis work for this case. For example, one central challenge is to ensure that the input graph, originally planar, maintain some planarity properties even after applying a sequence of contractions. This task is not very difficult if the contractions were performed in graphs, but when we have to process hypergraphs, maintaining planarity seems to be entirely *hopeless*. Still, we will show how to efficiently model the connectivity information of the hypergraph using the concept of shadow graphs that are unions of planar graphs.

The analysis is long, with many subtle fine points, and is presented in details in Sections III–VII.

Remark 3: While in our analysis we did not try to optimize the complexity of the H-freeness tester, focusing on the task of obtaining the query complexity of $O_{\varepsilon,H}(1)$, let us mention that in fact, with the analysis as it is now, without any optimization efforts, the complexity of our tester is doubly exponential in $|V(H)|/\varepsilon$.

Remark 4: While our main focus is on the random neighbor oracle model, it is straightforward to extend our testers and their analysis for H-freeness and for $\mathcal{H}$-freeness to the other two oracle access models presented in Section I-B. (However, our main result, the characterization of testable properties in planar graphs (and Theorem 12), cannot be extended to the other models (cf. Section I-C4), except the variant of the random neighbor oracle from Remark 2.)

3) Characterization of graph properties testable with one-sided error for planar/minor-free graphs: By combining the results sketched in Sections I-C1 and I-C2, the following characterization of graph properties testable with one-sided error (in the random neighbor oracle model) for general planar graphs and for minor-free graphs follows:

Theorem 5: A graph property $\mathcal{P}$ is testable with one-sided error in the random neighbor oracle model for planar graphs (and for minor-free graphs) if and only if $\mathcal{P}$ is semi-subgraph-free.

The proof of Theorem 5 follows immediately from our Theorem 12 (necessary condition) and Theorems 38 and 40 (sufficient condition).

One can read this characterization informally as follows: *A graph property $\mathcal{P}$ is testable with one-sided error in the random neighbor oracle model for planar graphs (or for minor-free graphs) if and only if $\mathcal{P}$ can be described as a property of testing forbidden subgraphs of constant size (the maximum size of any forbidden subgraph can be a function of $\mathcal{P}$ and ε).*

4) Remarks on the sensitivity and robustness of the oracle access models: While our tester for H-freeness (Section I-C2) is robust, the characterization presented in Theorem 5 is very sensitive to the oracle model. For example, it might be natural to consider a variant of our random neighbor oracle model to allow for every vertex to query for *distinct* random neighbors. That is, each time we call the random distinct neighbor query for a given vertex v, the oracle will return a neighbor of v chosen i.u.r. among all neighbors not returned earlier. One important feature of this model is that after $\deg(v)+1$ queries for a random distinct neighbor of vertex v, we are able to detect the degree $\deg(v)$ of vertex v in the input graph. This makes this model more powerful than our random neighbor oracle model, and in particular, it allows to test some properties that cannot be reduced to testing for forbidden subgraphs. For example, in that model one can test connectivity with $O(1/\varepsilon^3)$ queries and one-sided error (see, e.g., [18]). Indeed, if the input graph G is ε-far from being connected, then it is easy to see that G must have $\frac{1}{2}\varepsilon n$ connected components of size at most $\frac{2}{\varepsilon}$. Therefore, after randomly sampling $\frac{3}{\varepsilon}$ vertices, with probability at least $\frac{2}{3}$ one of the randomly sampled vertices will be in one of these small connected components. Since all vertices in this component must have degree at most $\frac{2}{\varepsilon}$, we can run BFS algorithm to explore the entire connected component with $O(1/\varepsilon^2)$ random distinct neighbor queries and verify that this connected component is indeed small, proving that the input graph is ε-far from being connected. This can be easily formalized to obtain a one-sided error tester for connectivity with query complexity $O(1/\varepsilon^3)$ in the random distinct neighbor oracle model. However, this task cannot be efficiently performed in our random neighbor oracle model (since we can never confirm with a finite number of queries a degree of a given vertex, even if its degree is constant, even if it is 1), and indeed, connectivity testing cannot be reduced to testing for a finite family of forbidden subgraphs and is not is a semi-subgraph-free graph property, even in planar graphs. (This is in contrast to other characterizations presented earlier in the literature, e.g., in [6], where the tester for the dense graphs model reduces to testing for forbidden *induced* subgraphs, giving a complete characterization of properties testable with one-sided error in terms of hereditary properties.) And so, even for planar graphs, *testing connectivity in the random neighbor oracle model is impossible with one-sided error*![2]

D. Organization of the paper

We begin in Section II with a formal analysis showing the necessary part of our characterization of testable properties, that any testable property is semi-subgraph-free (cf. Theorem 12 in Section II-D).

Then, in Sections III–VII, we present the *main technical contribution* of this paper, a complete analysis showing the sufficient part of our characterization of testable properties in planar graphs, that for any finite graph H, testing H-freeness is testable in planar graphs. The analysis here is split into several sections, with some auxiliary and technical results deferred to the full version of the paper [13]. We begin in Section III with an outline of the proof of testing H-freeness, focusing on connected H. Then, in Section IV, we present our tester and define our framework. Section V gives the first (and easiest) step in our transformation and show that any graph that is ε-far from H-free has a linear number of edge-disjoint copies of H. Then, in Section VI, we show how the contractions (cf. Section I-C2) can be performed in hypergraphs, to ensure existence of a sought set $\mathbb{Q}$ of edge-disjoint copies of H in which we can detect a copy of H. The analysis is then completed in Section VII. Finally, in Section VIII we discuss the extension to families of arbitrary finite graphs and in Section IX we discuss the extension to minor-free graphs.

[2]To see this, consider two *planar* graphs: a cycle C_n on n vertices, which is connected, and a perfect matching M_n on n vertices, which is ε-far from connected (for $\varepsilon < \frac{1}{2}$). Any tester should reject M_n with probability at least $\frac{2}{3}$. But at the same time, if we consider the tester on C_n (which must be accepted) then after performing q queries, with probability at least 2^{-q}, and so with positive probability, it will see only a subgraph of M_n. Therefore, since we consider one-sided error testers which must accept C_n, we conclude that no *one-sided error* tester *can reject* M_n.

Some final conclusions are in Section X.

II. Any testable property can be reduced to testing for forbidden subgraphs

In this section we provide a formal proof of the necessary (and easier) condition in our characterization, Theorem 12, that any one-sided-error testable property for arbitrary graphs can be reduced to testing for forbidden subgraphs of constant size (this claims holds for any finite family of graphs, not only for planar graphs). It should be noted that each graph in the family of forbidden graphs may have size depending on ε, the proximity parameter of the tester.

Our analysis critically relies on a recently developed in [9] canonical tester that shows that to test any testable graph property we can assume that the tester at hand is "oblivious" and works non-adaptively. This will allow us later to obtain a clean characterization of forbidden subgraphs for any given testable property $\mathcal{P}$.

A. Bounded-breadth bounded-depth graph exploration and bounded-discs

Our analysis relies on a random (BFS-like) bounded-breadth bounded-depth search, Bounded-BFS-Traverse below, an exploration algorithm similar to BFS of depth $\mathfrak{t}$. The algorithm runs from a given vertex a random BFS-like exploration of breadth $\mathfrak{d}$ and of depth $\mathfrak{t}$ using the random neighbor oracle (i.e., every vertex selects $\mathfrak{d}$ of its neighbors i.u.r. and recursively continues the process from them, until depth $\mathfrak{t}$ is reached). The main difference is that instead of visiting all neighbors of every vertex, like in the standard BFS algorithm, we *visit only* $\mathfrak{d}$ *neighbors chosen i.u.r.*, to limit the complexity of the search algorithm.

Bounded-BFS-Traverse $(G, v, \mathfrak{d}, \mathfrak{t})$:
- Let $L_0 = \{v\}$.
- For $\ell = 1$ to $\mathfrak{t}$ do:
  - ◇ Let $L_\ell = \emptyset$ and $\mathcal{E}_\ell = \emptyset$.
  - ◇ For every $u \in L_{\ell-1}$ do:
    - ○ Choose $\mathfrak{d}$ neighbors of u using $\mathfrak{d}$ *random neighbor queries*; call them Γ_u.
    - ○ Let $\mathcal{E}_u = \{(u,x) : x \in \Gamma_u\}$.
    - ○ Set $L_\ell = L_\ell \cup \Gamma_u$ and $\mathcal{E}_\ell = \mathcal{E}_\ell \cup \mathcal{E}_u$.
  - ◇ $L_\ell = L_\ell \setminus \bigcup_{i=0}^{\ell-1} L_i$.
- **Return** the subgraph of G induced by $\bigcup_{\ell=1}^{\mathfrak{t}} \mathcal{E}_\ell$.

We use the notion of bounded-breadth/depth search Bounded-BFS-Traverse to define bounded discs.

Definition 6: For given $\mathfrak{d}, \mathfrak{t} \in \mathbb{N}$, graph $G = (V, E)$, and vertex $v \in V$, a $(\mathfrak{d}, \mathfrak{t})$-*bounded disc* of v in G is any subgraph U of G that can be returned by Bounded-BFS-Traverse $(G, v, \mathfrak{d}, \mathfrak{t})$.

Vertex v is called a *root* of the $(\mathfrak{d}, \mathfrak{t})$-bounded disc U.

Let us observe that, assuming that $\mathfrak{d} \geq 2$, Bounded-BFS-Traverse $(G, v, \mathfrak{d}, \mathfrak{t})$ performs $\sum_{i=1}^{\mathfrak{t}} \mathfrak{d}^i \leq 2\mathfrak{d}^{\mathfrak{t}}$ queries to the input graphs. Accordingly, for $\mathfrak{d} \geq 2$, any $(\mathfrak{d}, \mathfrak{t})$-bounded disc has at most $\sum_{i=0}^{\mathfrak{t}} \mathfrak{d}^i \leq 2\mathfrak{d}^{\mathfrak{t}}$ vertices and at most $\sum_{i=1}^{\mathfrak{t}} \mathfrak{d}^i \leq 2\mathfrak{d}^{\mathfrak{t}}$ edges.

B. Rooted graphs, their basic properties, and semi-rooted-subgraph-freeness

In our analysis it will be sometimes useful to consider also *rooted* graphs, that is, graphs with some number of vertices distinguished as special vertices called *roots*. (For example, bounded discs from Definition 6 are rooted graphs.) To analyze similarities between rooted graphs, we will use the following definition.

Definition 7: **(Root-preserving isomorphism)** Let $Q = (V(Q), E(Q))$ and $Q' = (V(Q'), E(Q'))$ be two rooted graphs. A *root-preserving isomorphism* between Q and Q', denoted $Q \cong_r Q'$, is a bijection $f : V(Q) \to V(Q')$ such that u is the root of $V(Q)$ if and only if $f(u)$ is the root of $V(Q')$, and $(u, v) \in E(Q)$ if and only if $(f(u), f(v)) \in E(Q')$.

If $Q \cong_r Q'$, then we say that Q is *root-preserving isomorphic* to Q'.

We will extend this definition to compare a rooted graph with its occurrences (in a sense of root-preserving isomorphisms) in a large graph (which does not necessarily have to be rooted).

Definition 8: Let G be an undirected graph and let Q be a rooted graph. A *rooted copy of Q in G* is a subgraph U of G such that one can assign the roots to U so that there is a root-preserving isomorphism between Q and the rooted version of U. For an arbitrary set $\mathcal{Q}$ of rooted graphs, we say that *G is $\mathcal{Q}$-rooted-free* if for every $Q \in \mathcal{Q}$, there is no rooted copy of Q in G.

With these definitions, we are ready to present our auxiliary graph property notion.

Definition 9: **(Semi-rooted-subgraph-freeness)** A graph property $\mathcal{P}$ is *semi-rooted-subgraph-free* if for every ε, $0 < \varepsilon < 1$, and every $n \in \mathbb{N}$, there is a finite family $\mathcal{H}$ of finite graphs such that the following hold:

(i) any graph G satisfying $\mathcal{P}_n$ is $\mathcal{H}$-rooted-free, and
(ii) any graph G which is ε-far from satisfying $\mathcal{P}_n$, is not $\mathcal{H}$-rooted-free.

Similarly as in Definition 1, the notion of a family $\mathcal{H}$ of *finite graphs* means that every graph $H \in \mathcal{H}$ is finite, i.e., $|V(H)| = O_\varepsilon(1)$.

C. Modeling forbidden subgraphs in rooted graphs

While our analysis uses rooted graphs, their use is purely auxiliary because of the following simple fact.

Lemma 10: If a graph property $\mathcal{P}$ is semi-rooted-subgraph-free then $\mathcal{P}$ is also semi-subgraph-free.

Proof. This follows easily from the definitions of semi-rooted-subgraph-free and semi-subgraph-free properties. For any rooted graph H, let $\overline{H}$ denote the same graph with removed roots (that is, we remove the labels defining the roots); similarly, for any family $\mathcal{H}$ of rooted graphs, let $\overline{\mathcal{H}} = \{\overline{H} : H \in \mathcal{H}\}$. Then we claim that for any graph G be an arbitrary graph and any family $\mathcal{H}$ of rooted graphs,

(a) if G is $\mathcal{H}$-rooted-free then G is also $\overline{\mathcal{H}}$-free, and
(b) if G is not $\mathcal{H}$-rooted-free, then G is also not $\overline{\mathcal{H}}$-free.

Indeed, to see part *(a)*, suppose, by contradiction, that G is not $\overline{\mathcal{H}}$-free, that is, there is $\overline{H}$ with $H \in \mathcal{H}$ such that $\overline{H}$ is a subgraph of G. But then G has a rooted copy of H, since we can take the roots of H and assign them to $\overline{H}$, so that there is a root-preserving isomorphism between H and the rooted version of $\overline{H}$. Since G has a rooted copy of H, we conclude that G is not $\mathcal{H}$-rooted-free, which is contradiction.

To see part *(b)*, suppose, by contradiction, that G is $\overline{\mathcal{H}}$-free, that is, there is no $\overline{H} \in \overline{\mathcal{H}}$ such that G has a copy of $\overline{H}$. But then, clearly, G is $\mathcal{H}$-rooted-free, since otherwise, there would be $H \in \mathcal{H}$ such that G had a rooted copy of H, which would imply that $\overline{H}$ was a subgraph G; contradiction.

Now, we are ready to complete the proof of Lemma 10. By Definition 9, if $\mathcal{P}$ is semi-rooted-subgraph-free then there exists a finite family $\mathcal{H}$ of finite rooted graphs such that (i) any graph G satisfying $\mathcal{P}$ is $\mathcal{H}$-rooted-free, and (ii) any graph G which is ε-far from satisfying $\mathcal{P}$, is not $\mathcal{H}$-rooted-free. If we combine these properties with our claim above, then we obtain that for the finite family of finite graphs $\overline{\mathcal{H}} = \{\overline{H} : H \in \mathcal{H}\}$,

(i') any graph G satisfying $\mathcal{P}$ is $\mathcal{H}$-rooted-free, and thus (by *(a)*) also $\overline{\mathcal{H}}$-free, and
(ii') any graph G which is ε-far from satisfying $\mathcal{P}$, is not $\mathcal{H}$-rooted-free, and thus (by *(b)*) also not $\overline{\mathcal{H}}$-free.

Therefore $\mathcal{P}$ is semi-subgraph-free (cf. Definition 1). ■

D. Canonical testers and reduction to testing for forbidden subgraphs

Next, our analysis follows the framework described in Section I-C1. We rely on the following Theorem 3.6 from [9] describing a *canonical way of designing any tester in the random neighbor oracle model.*

*Theorem 11 (**Canonical tester [9]**):* Let $\mathcal{P} = (\mathcal{P}_n)_{n\in\mathbb{N}}$ be a graph property that can be tested in the random neighbor oracle model with query complexity $q = q(\varepsilon)$ and error probability at most $\frac{1}{3}$. Then for every ε, there exists $q' = \Theta(q)$, and an infinite sequence $\mathcal{Q} = (\mathcal{Q}_n)_{n\in\mathbb{N}}$ such that for every $n \in \mathbb{N}$,

- $\mathcal{Q}_n$ is a set of rooted graphs such that each $Q \in \mathcal{Q}_n$ is the union of q' many (q', q')-bounded discs;
- the property $\mathcal{P}_n$ on n-vertex graphs can be tested with error probability at most $\frac{1}{3}$ by the following canonical tester (with query complexity $q^{O(q)}$):
  - ◇ sample a set (possibly, a multiset) S of q' vertices chosen i.u.r.;
  - ◇ for each sampled vertex v, run Bounded-BFS-Traverse (G, v, q', q') to get a (q', q')-bounded disc U_v;
  - ◇ reject if and only if there exists a root-preserving isomorphism between the union of the explored (q', q')-bounded discs and some element $Q \in \mathcal{Q}_n$, that is, there is $Q \in \mathcal{Q}_n$ with $\bigcup_{v\in S} U_v \cong_r Q$.

Furthermore, if $\mathcal{P} = (\mathcal{P}_n)_{n\in\mathbb{N}}$ can be tested in the random neighbor oracle model with query complexity $q(\varepsilon)$ with one-sided error, then the resulting canonical tester for $\mathcal{P}$ has one-sided error too.

Theorem 11 from [9] shows that without loss of generality, we can assume that any testable graph property can be tested by a canonical tester with constant query complexity. With Theorem 11, Lemma 10, and Definitions 1 and 9 at hand, we are now ready to present the main result of this section.

Theorem 12: If a graph property $\mathcal{P}$ is testable with one-sided error in the random neighbor oracle model then $\mathcal{P}$ is semi-subgraph-free.

Proof. First, notice that thanks to Lemma 10, it is enough to show that if a graph property $\mathcal{P}$ is testable with one-sided error in the random neighbor oracle model then $\mathcal{P}$ is semi-*rooted*-subgraph-free (cf. Definition 9).

Let us fix $n \in \mathbb{N}$ and ε, and suppose that $\mathcal{P}_n$ is a graph property on n-vertex graphs that can be tested in the random neighbor oracle model with query complexity $q(\varepsilon)$ and one-sided error. By Theorem 11 from [9], we can assume that $\mathcal{P}_n$ is tested by a canonical tester $\mathfrak{T}$ that satisfies the conditions of Theorem 11. In particular, let $\mathcal{Q}_n$ be the family of forbidden rooted graphs for $\mathcal{P}_n$ (union of q' many (q', q')-bounded discs) whose existence follows from Theorem 11. We will show that so defined family of rooted graphs satisfies the conditions in Definition 9, proving that $\mathcal{P}$ is semi-subgraph-free.

Let us first notice that each rooted graph $\mathcal{Q}_n$ has at most $2(q')^{q'}$ vertices and at most $2(q')^{q'}$ edges, and so $\mathcal{Q}_n$ is a finite family of finite rooted graphs.

Let us next show item *(i)* of Definition 9, that any n-vertex graph G satisfying $\mathcal{P}_n$ is $\mathcal{Q}_n$-rooted-free (cf. Definition 8). The proof is by contradiction and so suppose that there is a graph G satisfying $\mathcal{P}_n$ which contains a rooted copy of $Q \in \mathcal{Q}_n$. Then, with a positive probability the canonical tester $\mathfrak{T}$ on G will take that copy of $Q \in \mathcal{Q}_n$, and by the definition, it will reject G. This means that the tester has a nonzero probability of rejecting G, contradicting our assumption that the tester $\mathfrak{T}$ is one-sided.

Now, we want to prove item *(ii)* of Definition 9. Let G be an n-vertex graph that is ε-far from satisfying $\mathcal{P}_n$. Any tester for $\mathcal{P}_n$ should reject G with nonzero probability. By definition of our canonical tester $\mathfrak{T}$, G is rejected by $\mathfrak{T}$ only if G contains a rooted subgraph U such that if the tester $\mathfrak{T}$ gets U from the oracle, then $U \cong_r Q$. By definition of $\mathfrak{T}$ and $\mathcal{Q}_n$ this means that $Q \in \mathcal{Q}_n$, which proves item *(ii)* of Definition 9.

We have shown that if a graph property $\mathcal{P}$ is testable with one-sided error in the random neighbor oracle model then $\mathcal{P}$ is semi-rooted-subgraph-free. By Lemma 10, this yields that $\mathcal{P}$ is semi-subgraph-free, completing the proof. ■

Remark 13: While Theorem 12 is presented for any general graphs, it is straightforward to extend it to hold also for infinite classes of graphs, for example, for planar graphs, or for the family of minor-closed graphs.

E. Uniform characterization using oblivious testers and forbidden subgraphs

While Theorem 11 from [9] allows to simplify the analysis of testable properties, the analysis as in Theorem 12 obtains non-uniform testers, in the sense of the dependency on n. We could make our result uniform by considering a special class of uniform testers, which we call *oblivious testers*, that capture the essence of testers of testable properties in the flavor of Theorem 11 (see [6] for a similar notion in the context of testing dense graphs). This characterization is discussed in details in the full version of the paper [13].

III. TESTING H-FREENESS: HIGH-LEVEL VIEW

We begin our analysis with fixing an arbitrary finite, *connected*, undirected, simple graph $H = (V(E), E(H))$.[3]

Our tester of H-freeness relies on a simple graph exploration. We first describe our algorithm for testing H-freeness of planar graphs with arbitrary degrees and provide the high level structure of its analysis. We defer most of technical details to Sections IV– VII and the full version of the paper [13].

Our algorithm relies on a random bounded-breadth bounded-depth search, Random-Traverse below, which uses Bounded-BFS-Traverse $(G, v, \mathfrak{d}, \mathfrak{t})$ from Section II-A. (Let us remind, cf. page 6, that Bounded-BFS-Traverse $(G, v, \mathfrak{d}, \mathfrak{t})$ is similar to BFS of depth $\mathfrak{t}$ starting at vertex v, though instead of visiting all neighbors of every vertex, one visits only $\mathfrak{d}$ neighbors chosen i.u.r., to limit the complexity of the algorithm.)

Random-Traverse $(G, \mathfrak{d}, \mathfrak{t})$:
- Pick a random vertex $v \in V$ i.u.r. and run Bounded-BFS-Traverse $(G, v, \mathfrak{d}, \mathfrak{t})$.

Our tester Random-Exploration runs $f(\varepsilon, H)$ times our search algorithm Random-Traverse with parameters $\mathfrak{d} = h(\varepsilon, H)$, $\mathfrak{t} = g(\varepsilon, H)$, each time checking if the graph induced by the visited edges contains a copy of H, or does not. The algorithm accepts G as H-free if and only if all calls found no copy of H in G.

Tester: Random-Exploration (G, H, ε):
(with three implicit parameters, integer functions f, g, h)
- Repeat $f(\varepsilon, H)$ times:
  - ◇ Run Random-Traverse $(G, h(\varepsilon, H), g(\varepsilon, H))$ and let $\mathcal{E}$ be the resulted set of edges.
  - ◇ If the subgraph of G induced by the edges $\mathcal{E}$ contains a copy of H, then **reject**.
- If every subgraph explored is H-free, then **accept**.

The following main theorem describes key properties of our tester.

Theorem 14: Let H be connected. There are positive functions f, g, h, such that for any planar graph G:
- if G is H-free, then Random-Exploration(G, H, ε) accepts G, and
- if G is ε-far from H-free, then Random-Exploration(G, H, ε) rejects G with probability at least 0.99.

[3] While our analysis here assumes that H is connected, this is clearly not required for the main result. If H is disconnected then with the coloring trick (cf. Section IV-A1), one could have identical analysis and consider all connected components one by one, extending the results to arbitrary, not necessarily connected H. We will discuss this in details in Section VIII.

It is obvious that the first claim holds: if G is H-free, then so is every subgraph of G, and therefore Random-Exploration always accepts. Therefore, to prove our main result, Theorem 14, it suffices to show that if G is ε-far from H-free, then Random-Exploration rejects G with probability at least 0.99. In view of that, from now on, we assume that the input graph G is ε-far from H-free for some constant $\varepsilon > 0$.

We note that it is enough to show that a *single* instance of the random bounded-breadth bounded-depth search (Random-Traverse) of breadth $O_{\varepsilon,H}(1)$ and depth $O_{\varepsilon,H}(1)$ finds a copy of H with probability $\Omega_{\varepsilon,H}(1)$. Indeed, for any functions f, g, and h, if Random-Traverse $(G, \mathfrak{d}, \mathfrak{t})$ with $h(\varepsilon, H) = O_{\varepsilon,H}(1)$ and $g(\varepsilon, H) = O_{\varepsilon,H}(1)$ finds a copy of H with probability at least $5/f(\varepsilon, H) = \Omega_{\varepsilon,H}(1)$, then this implies that $f(\varepsilon, H) = O_{\varepsilon,H}(1)$ independent calls to Random-Traverse $(G, \mathfrak{d}, \mathfrak{t})$ detect at least one copy of H with probability at least $1-(1-5/f(\varepsilon,H))^{f(\varepsilon,H)} \geq 1-e^{-5} \geq 0.99$. Therefore, in the remainder of the paper, we analyze the following algorithm Tester$(G, H, \mathfrak{d}, \mathfrak{t})$.

Tester $(G, H, \mathfrak{d}, \mathfrak{t})$:
- Run Random-Traverse $(G, \mathfrak{d}, \mathfrak{t})$ and let $\mathcal{E}$ be the resulted set of edges.
- If the subgraph of G induced by the edges $\mathcal{E}$ contains a copy of H, then **reject**.
- If not, then **accept**.

We will show the following central technical theorem.

Theorem 15: Let H be a connected undirected graph. There are positive functions $\mathfrak{d} = \mathfrak{d}(\varepsilon, H) = O_{\varepsilon,H}(1)$ and $\mathfrak{t} = \mathfrak{t}(\varepsilon, H) = O_{\varepsilon,H}(1)$ such that for any planar graph G that is ε-far from H-free, Tester$(G, H, \mathfrak{d}, \mathfrak{t})$ finds a copy of H with probability $\Omega_{\varepsilon,H}(1)$. The query complexity of Tester$(G, H, \mathfrak{d}, \mathfrak{t})$ is $O(\mathfrak{d}^{\mathfrak{t}}) = O_{\varepsilon,H}(1)$.

Since by our discussion above Theorem 15 yields Theorem 14, we will focus on proving Theorem 15. We also notice that the query complexity of the tester follows directly from its definition, and so we will concentrate on showing that for $\mathfrak{d} = O_{\varepsilon,H}(1)$ and $\mathfrak{t} = O_{\varepsilon,H}(1)$, Tester$(G, H, \mathfrak{d}, \mathfrak{t})$ finds a copy of H with probability $\Omega_{\varepsilon,H}(1)$.

A. Outline of the proof of testing H-freeness

In this subsection we outline the key ideas behind our proof of testing H-freeness. Since the proof is long and complex, we will give here mostly some underlying intuitions, leaving the details to Sections IV–VII.

By our discussion above, it suffices to focus on the case when the input graph G is ε-far from H-free. Our analysis relies on the following result (shown in Lemma 17) that every simple planar graph G that is ε-far from H-free has a subgraph $\mathbb{G}$ satisfying the following:

(a) if Tester$(\mathbb{G}, H, \mathfrak{d}, \mathfrak{t})$ finds a copy of H in $\mathbb{G}$ with probability $\Omega_{\varepsilon,H}(1)$, then Tester$(G, H, \mathfrak{d}, \mathfrak{t})$ finds a copy of H in G with probability $\Omega_{\varepsilon,H}(1)$, and
(b) Tester$(\mathbb{G}, H, \mathfrak{d}, \mathfrak{t})$ finds a copy of H in $\mathbb{G}$ with probability $\Omega_{\varepsilon,H}(1)$.

Our first (and easy) step towards proving this property is to show that G contains a linear number of edge-disjoint copies of H (see Lemma 20). This follows by iteratively removing copies of H and observing that by the definition of being ε-far from H-free, we have to remove εn edges to make G free of copies of H. In the following we will use $\mathbb{Q}$ to denote a set (of linear size) of edge-disjoint copies of H in G. We continue by showing that given $\mathbb{Q}$, we can compute a subset $\mathbb{Q}'$ of linear size such that the graph $G[\mathbb{Q}']$ (subgraph of G on vertex set V and with edge set being the union of the edges of the subgraphs in $\mathbb{Q}'$) satisfies the first property above. The proof essentially shows that one can remove copies from $\mathbb{Q}'$ until every vertex in $G[\mathbb{Q}']$ has degree either 0 or a small positive constant times its degree in G.

Next, we would like to define a sequence of sets $\mathbb{Q} = \mathbb{Q}_0 \supseteq \mathbb{Q}_1 \supseteq \cdots \supseteq \mathbb{Q}_{|V(H)|}$ with associated hypergraphs with the following interpretation. The hyperedges will be labelled in such a way that we are able to recover the set $\mathbb{Q}_i$ from it. We will use hyperedges to replace certain subgraphs of H and their corresponding part in G.

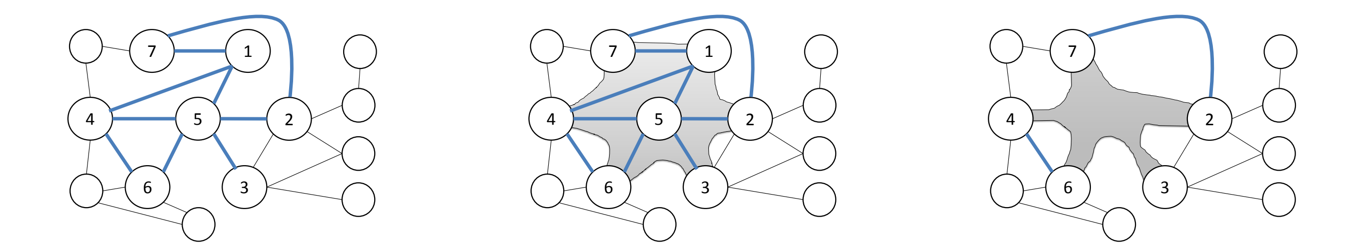
Hyperedges: We will now describe the use of hyperedges as replacements for copies of subgraphs of H in G. Let G^* be a subgraph of G that has a copy of H. Consider a subgraph H_1 of H and let $u_1, \ldots, u_\ell$ be the vertices in the copy of H_1 in G^* that separate $G \setminus H_1$ from H_1, so that (cf. Figure 1):

(a) every vertex from $\{u_1, \ldots, u_\ell\}$ is adjacent in G^* to some vertex $H_1 \setminus \{u_1, \ldots, u_\ell\}$,
(b) every vertex in $H_1 \setminus \{u_1, \ldots, u_\ell\}$ is adjacent in G^* only to vertices from H_1, and
(c) $\{u_1, \ldots, u_\ell\}$ forms an independent set in H_1.

Then, we can construct a gadget to represent that copy of H_1 by removing from H_1 all vertices and edges from $H_1 \setminus \{u_1, \ldots, u_\ell\}$ and replacing them by a single *hyperedge* $\{u_1, \ldots, u_\ell\}$.

We will encode the structural information of the subgraph replaced by the hyperedge in a label, so that it may happen that we have parallel hyperedges with different labels. In addition to the above structural role we recall from the previous section that the idea of hyperedges was to encode that whenever (a hypergraph version of) Random-Traverse enters the hyperedge then it will reach all its vertices. Our final goal will be to construct a hypergraph that only consists of selfloops, so that we can argue easily that our tester finds a copy of H by finding a corresponding set of labelled selfloops.

Vertex coloring: A major difficultly in applying our approach is to find subgraphs that can be replaced. One way to simplify this question is to color both the vertices of H and the vertices of G with $|V(H)|$ colors, such that every vertex of H receives a distinct color and every copy of H in $\mathbb{Q}$ has the same coloring as H. We show in Lemma 20 that there is a coloring χ of G and H such that G contains a set $\mathbb{Q}$ containing a linear number of such edge-disjoint colored copies of H. An important feature of this coloring, which will be very useful in finding vertices that can be replaced by hyperedges, is that *every vertex has the same role in all subgraphs from* $\mathbb{Q}$ it is contained in.



(a) (b) (c)

Fig. 1. (a) Consider a part of the input graph G with numbered vertices corresponding to the colored vertices in a copy of H in G and thick edges corresponding to the edges in that copy of H. (b) We have a subgraph H_1 of H consisting of the vertices and edges marked by the grey area, with vertices $\{2,3,4,6,7\}$ separating H_1 from the rest of G. (c) The gadget obtained by removing internal vertices $\{1,5\}$ and replacing H_1 by a hyperedge connecting vertices $\{2,3,4,6,7\}$.

Getting from $\mathbb{Q}_i$ to $\mathbb{Q}_{i+1}$: Next we describe how we move from the set $\mathbb{Q}_i$ to $\mathbb{Q}_{i+1}$. This is the main step in our reduction and it will be partitioned in a number of substeps. We start with an overview. In each round we perform the following high level process:

- Select a vertex $v_i \in V(H)$.
- Simultaneously, contract every vertex $u \in V(\mathcal{H}_i(\mathbb{Q}_i))$ with $\chi(u) = \chi(v_i)$ as follows:
  - ◇ for every colored copy $\mathfrak{h}$ of H in $\mathbb{Q}_{i+1}$ that contains vertex u:
    - ○ add a new hyperedge consisting of vertices in $\mathcal{N}_i^{\mathfrak{h}}\langle u\rangle$, where $\mathcal{N}_i^{\mathfrak{h}}\langle u\rangle$ is the set of neighbors of u in $\mathfrak{h}$ (in the corresponding hypergraph) other than u (that is, $u \notin \mathcal{N}_i^{\mathfrak{h}}\langle u\rangle$);
  - ◇ remove vertex u (with all incident edges from $\mathcal{H}_i(\mathbb{Q}_i)$).

We remark that our algorithm above ensures that no neighboring vertices are contracted since the coloring χ has no monochromatic edges. This follows from the fact that every edge in $G[\mathbb{Q}]$ belongs to some copy of H and the coloring of H has no monochromatic edge. Thus, we can perform the contractions independently.

In our construction we will require that the contracted vertices additionally satisfy some stronger properties. This is to maintain (approximately) some basic properties of planar graphs.

- We want to ensure that all contractions in $\mathcal{H}_i(\mathbb{Q}_i)$ corresponding to the contraction of v_i are *consistent*, that is, the contraction of u is the same in every colored copy of H that contains u (that is, for every vertex u in with $\chi(u) = \chi(v_i)$, for any two colored copies $\mathfrak{h}_1, \mathfrak{h}_2$ of H in $\mathbb{Q}_{i+1}$ containing vertex u, we have $\mathcal{N}_i^{\mathfrak{h}_1}\langle u\rangle = \mathcal{N}_i^{\mathfrak{h}_2}\langle u\rangle$).

The required property is captured in the following definition (see also Definition 26).

Definition 16: **(Safe vertices)** Let $\mathbb{Q}_i$ be a set of edge-disjoint colored copies of H in G and let $\mathbb{Q} \subseteq \mathbb{Q}_i$. We call a vertex u *safe* if for all colored copies $\mathfrak{h} \in \mathbb{Q}$ of H that contain u, the sets $\mathcal{N}_i^{\mathfrak{h}}\langle u\rangle$ are the same.

Finding safe vertices: Our next challenge is to show that we can find many (a linear number) safe vertices of the same color. In order to do so, we will delete elements from the current set $\mathbb{Q}_i$ in a controlled way until we can guarantee that many safe vertices of the same color exist. An important concept that we define here is that of a *shadow graph*. A shadow graph is a union of $|V(H)|$ planar graphs and it models the neighborhood relation of our hypergraph, such that two vertices are adjacent in the shadow graph if and only if they belong to the same edge in the hypergraph. The main use of shadow graphs is to show in the upcoming construction that our hypergraph still satisfies some near-planar properties that will be useful in the analysis. The concept of shadow graphs and the proof of their existence is one of the main new ideas in this paper.

Using the existence of shadow graphs, we can properly implement the process of contractions via hyperedges, proceed similarly as in an earlier paper about testing of bipartiteness in planar graphs [11], where the shadow graphs guarantee that we still approximately satisfy the properties of planar graphs that were used the previous paper [11]: We first prove that we can construct a subset $\mathbb{Q}$ of $\mathbb{Q}_i$ of linear size such that every copy of H in $\mathbb{Q}$ has a vertex of constant degree in $G[\mathbb{Q}]$. Then we use this claim in the proof of Lemma 32 to show how to construct a subset $\mathbb{Q}^*$ of $\mathbb{Q}_i$ such that every copy of H in $\mathbb{Q}^*$ contains a safe vertex.

Wrapping things up: What remains to do is to prove that our construction satisfies the second required property of our tester:

> $\otimes$ Tester$(G[\mathbb{Q}], H, \mathfrak{d}, \mathfrak{t})$ finds a copy of H in $G[\mathbb{Q}]$ with probability $\Omega_{\varepsilon,H}(1)$.

We define $\mathbb{Q}$ to be the set $\mathbb{Q}_{|V(H)|}$ obtained in the final round of our reduction.

We will then prove $\otimes$ by showing two central properties (proven in Claims 36 and 37), which informally say that after all contractions $\mathcal{M}_{|V(H)|}$ only consists of selfloops, which can easily be found (Claim 36) and the proof of Claim 37 formalizes our idea that if our random walk enters a hyperedge in $\mathcal{H}_{i+1}(\mathbb{Q})$ then we perform with constant probability the same operation in $\mathcal{H}_{i+1}(\mathbb{Q})$ in two steps of our randomized process. Combining the results with our previous considerations yields our main statement: H-freeness in planar graphs is constant query-time testable.

IV. Analysis of Tester when G is ε-far from H-free

Because of the arguments from the previous section, the remainder of the paper deals with the main technical challenge of our result: proving Theorem 15 that in any simple planar graph $G = (V, E)$ that is ε-far from H-free, our algorithm Tester finds with sufficient probability a copy of H.

Our analysis relies on the following lemma showing the existence of a special subgraph $\mathbb{G}$ of G:

Lemma 17: For every $\varepsilon \in (0, 1)$, there are $\mathfrak{d} = \mathfrak{d}(\varepsilon, H)$ and $\mathfrak{t} = \mathfrak{t}(\varepsilon, H)$, such that for every simple planar graph $G = (V, E)$ that is ε-far from H-free, there is a subgraph $\mathbb{G}$ of G with the following properties:

(a) if Tester$(\mathbb{G}, H, \mathfrak{d}, \mathfrak{t})$ finds a copy of H in $\mathbb{G}$ with probability $\Omega_{\varepsilon,H}(1)$, then Tester$(G, H, \mathfrak{d}, \mathfrak{t})$ finds a copy of H in G with probability $\Omega_{\varepsilon,H}(1)$, and

(b) Tester$(\mathbb{G}, H, \mathfrak{d}, \mathfrak{t})$ finds a copy of H in $\mathbb{G}$ with probability $\Omega_{\varepsilon,H}(1)$.

Observe that if such a subgraph $\mathbb{G}$ as promised in Lemma 17 always exists, then these properties immediately imply that Tester$(G, H, \mathfrak{d}, \mathfrak{t})$ finds a copy of H in G with probability $\Omega_{\varepsilon,H}(1)$ and therefore, by the discussion above, Theorems 14 and 15 follow.

In order to prove Lemma 17, we will show that for any simple planar graph G that is ε-far from H-free, there exists a set $\mathbb{Q}$ of edge-disjoint copies of H in G for which $G[\mathbb{Q}]$, the subgraph of G induced by the edges of $\mathbb{Q}$, satisfies the properties of graph $\mathbb{G}$ in Lemma 17. The construction of the set $\mathbb{Q}$ and the analysis of its properties form the main technical contribution of our paper. While part (a) in Lemma 17 is rather easy to achieve and to analyze (thanks to Lemma 18 in Section IV-A2), the main challenge of our construction is in ensuring part (b) in Lemma 17. For that, we use a rather elaborate construction to gradually find a sequence $\mathbb{Q}_1 \supseteq \mathbb{Q}_2 \supseteq \cdots \supseteq \mathbb{Q}_{|V(H)|}$ of sets of edge-disjoint copies of H in G, with $|\mathbb{Q}_{|V(H)|}| = \Omega_{\varepsilon,H}(|V|)$, such that the final set $\mathbb{Q}_{|V(H)|}$ is the set $\mathbb{Q}$ that defines $\mathbb{G} = G[\mathbb{Q}_{|V(H)|}]$ in Lemma 17.

The construction of the sequence $\mathbb{Q}_1 \supseteq \mathbb{Q}_2 \supseteq \cdots \supseteq \mathbb{Q}_{|V(H)|}$ of sets of edge-disjoint copies of H in G, with $|\mathbb{Q}_{|V(H)|}| = \Omega_{\varepsilon,H}(|V|)$, for which we could easily argue that Tester$(G[\mathbb{Q}_{|V(H)|}], H, \mathfrak{d}, \mathfrak{t})$ finds a copy of H in $G[\mathbb{Q}_{|V(H)|}]$ with probability $\Omega_{\varepsilon,H}(1)$, is the most challenging and technical contribution of our paper. We begin with a simple construction of $\mathbb{Q}_1$ which is a set of $\Omega_{\varepsilon,H}(|V|)$ edge-disjoint copies of H in G (cf. Lemma 20). Then our construction is iterative: we design a reduction that takes set $\mathbb{Q}_i$ of $\Omega_{\varepsilon,H}(n)$ edge-disjoint copies of H and we construct from it another set $\mathbb{Q}_{i+1} \subseteq \mathbb{Q}_i$ with $|\mathbb{Q}_{i+1}| = \Omega_{\varepsilon,H}(|\mathbb{Q}_i|)$ for which we simplify the structure of $G[\mathbb{Q}_{i+1}]$ with respect to that of $G[\mathbb{Q}_i]$. To guide our process, we associate with each $\mathbb{Q}_i$ a certain *hypergraph* $\mathcal{H}_i(\mathbb{Q}_i)$ that is constructed from $\mathbb{Q}_i$ by contracting vertices of H in a specific, consistent way (cf. Section VI-C). *The purpose of $\mathcal{H}_i(\mathbb{Q}_i)$ is to model the copies of H by a hypergraph on a smaller number of vertices, by contracting vertices (and incident edges) which are known to be visited by Random-Traverse via other means.* We will construct a sequence of hypergraphs $\mathcal{H}_1(\mathbb{Q}_1), \mathcal{H}_2(\mathbb{Q}_2), \dots, \mathcal{H}_{|V(H)|}(\mathbb{Q}_{|V(H)|})$ that correspond to sets $\mathbb{Q}_1, \mathbb{Q}_2, \dots, \mathbb{Q}_{|V(H)|}$, and a sequence of hypergraphs $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_{|V(H)|}$ that are "shrunk" copies of H, each $\mathcal{M}_i$ with $|V(H)| - i + 1$ vertices, such that, informally, for our algorithm of selecting $\mathbb{Q}_1, \mathbb{Q}_2, \dots, \mathbb{Q}_{|V(H)|}$, the following conditions holds:

- the probability of finding by Random-Traverse a copy of H in $G[\mathbb{Q}_1]$ is the same as the probability of finding by Random-Traverse a copy of $\mathcal{M}_1$ in $\mathcal{H}_1(\mathbb{Q}_1)$,
- the probability of finding by Random-Traverse a copy of $\mathcal{M}_{i+1}$ in $\mathcal{H}_{i+1}(\mathbb{Q}_{i+1})$ is similar to the probability of finding by Random-Traverse a copy of $\mathcal{M}_i$ in $\mathcal{H}_i(\mathbb{Q}_i)$, and
- using the fact that $\mathcal{M}_{|V(H)|}$ has a single vertex, one can easily estimate the probability of finding by Random-Traverse a copy of $\mathcal{M}_{|V(H)|}$ in $\mathcal{H}_{|V(H)|}(\mathbb{Q}_{|V(H)|})$.

With these three properties at hand, the main theorem will follow.

One central feature of our analysis via the study of hypergraphs is to ensure that the underlying hypergraphs have some basic planar graphs-like properties. (In particular, informally, in our analysis we would like to argue that there is always a constant fraction of low-degree vertices.) While we do not have a useful characterization of planar hypergraphs, we will be able to model some planarity-like properties of the hypergraphs using some special graph reduction (via *shadow graphs*), see Lemma 29 (and for more details, see the full version of the paper [13]).

In the following sections we will develop this framework in details, finalizing it in Section VII that proves the desired properties above.

A. Auxiliary technical tools

We begin with three auxiliary tools in our analysis, the study of the problem of finding *colored* copies of H in G (Section IV-A1), a reduction simplifying condition (a) of Lemma 17 (Section IV-A2), and extension of the testing and graph exploration framework to hypergraphs (Section IV-A3).

1) Auxiliary tools: Finding colored copies *of H in G:* To simplify the analysis, we will consider colored copies of H in G. Let us *color* all vertices of H using $|V(H)|$ colors, one color for each vertex (without loss of generality, the colors are $\{1, 2, \dots, |V(H)|\}$). While the coloring is not needed by the algorithm, it will simplify the analysis. With this in mind, instead of showing that our algorithm Tester finds with sufficient probability a copy of H, we will show (cf. Lemma 20) that there is a coloring χ of vertices of G such that Tester finds (with sufficient probability) a colored copy of H, that is, a copy of H in G with colors of the vertices in the copy consistent with the coloring χ. (While this statement sounds trivial, since once we found a copy of H in G we can always color vertices of G to be consistent with the coloring of H, the colors will be helpful in our analysis.) Therefore, from now on, whenever we will aim to find a copy of H we will mean to find a colored copy of H consistent with given coloring χ.

Let us notice one immediate implication of this assumption: if $\mathbb{Q}_i$ and χ are fixed, then one can think about every edge e as a *labeled edge*, since the colors of its endpoints define a unique edge in H that e corresponds too. We will use this property implicitly throughout the paper, without mentioning it anymore.

2) Auxiliary tools: Simplifying condition (a) of Lemma 17: (via edge-disjoint copies of H): We show that one can simplify condition (a) of Lemma 17 for the special case when the subgraph $\mathbb{G}$ of G is a union of a linear number of edge-disjoined colored copies of H (a similar approach has been also used in [11]). That is, if there is a graph $G[\mathbb{Q}]$ with a *linear number of edge-disjoint colored copies of H*, then Lemma 18 shows that there is always a subset $\mathbb{Q}' \subseteq \mathbb{Q}$ with cardinality $|\mathbb{Q}'| = \Omega_{\varepsilon,H}(|\mathbb{Q}|)$ such that the graph $G[\mathbb{Q}']$ satisfies property (a).

Lemma 18: ***(Transformation to obtain property (a))*** Let $G = (V, E)$ be a simple planar graph. Let $\mathbb{Q}$ be a set of $\Omega_{\varepsilon,H}(|V|)$ edge-disjoint colored copies of H in G. Then there exists a subset $\mathbb{Q}' \subseteq \mathbb{Q}$, $|\mathbb{Q}'| = \Omega_{\varepsilon,H}(|V|)$, such that the graph $G[\mathbb{Q}']$ satisfies condition (a) of Lemma 17.

The proof of Lemma 18, as a natural extension of the approach from [11], is deferred to the full version of the paper [13].

3) Traversing hypergraphs and testing hypergraph $\mathcal{M}$-freeness: In Section IV, we described two central algorithms used for testing H-freeness: Random-Traverse, and Tester. Both these algorithms were presented in a form required to test H-freeness in a graph. However, in our transformations we will apply the same algorithms to hypergraphs, to test whether a hypergraph $\mathcal{H}$ (in a form of $\mathcal{H}_i(\mathbb{Q}_i)$, as defined in Section VI-C) is $\mathcal{M}$-free, where $\mathcal{M}$ is a fixed hypergraph (which in our applications will be $\mathcal{M}_i$, as defined in Section VI-B). While the modifications are rather straightforward, for the sake of completeness, we will describe below these algorithms to be run on a hypergraph. Furthermore, in our algorithms for hypergraphs we will have one additional parameter, a *representative function* $\mathfrak{Rep} : V \to V$, which describes the way how the edges have been contracted (cf. Definition 34). The idea behind the representative function $\mathfrak{Rep}$ is that any vertex u that either is in the hypergraph $\mathcal{H}$ or which does not belong to any set of copies of H has $\mathfrak{Rep}(u) = u$, but any other vertex u from G that has been contracted and now does not appear in $\mathcal{H}$, has $\mathfrak{Rep}(u)$ equal to its representative in $\mathcal{H}$. In the latter case, the intuition is that the representative is a vertex in $\mathcal{H}$ that with probability $\Omega_{\varepsilon,H}(1)$ can be reached from u in $O_{\varepsilon,H}(1)$ steps, if Random-Traverse (run in G) started at u.

Remark 19: Let us remark that in HTester and Random-HTraverse below we use the input graph G implicitly, since in Random-HTraverse we directly refer here to the set V, which is the vertex set of G, and we do so indirectly via the use of $\mathfrak{Rep}$, whose domain and range are V. Further, in our applications we will always have that $V(\mathcal{H}) \subseteq V$.

Random-HTraverse $(\mathcal{H}, \mathfrak{Rep}, \mathfrak{d}, \mathfrak{t})$:

- Pick a vertex $v \in V$ i.u.r., and let $L_0 = \{\mathfrak{Rep}(v)\}$ (i.e., L_0 has a randomly selected vertex, such that any $u \in V$ is chosen with probability $\frac{|\mathfrak{Rep}^{(-1)}(u)|}{|V|}$).
- If v is a vertex of $\mathcal{H}$ then for $\ell = 1$ to $\mathfrak{t}$ do:
  - ◇ Let $L_\ell = \emptyset$ and $\mathcal{E}_\ell = \emptyset$.
  - ◇ For every $u \in L_{\ell-1}$ do:
    - ○ Choose $\mathfrak{d}$ edges incident to u in $\mathcal{H}$ i.u.r.; call them $\mathcal{E}_{\ell,u}$.
    - ○ Let Γ_u be the set of vertices in $\mathcal{E}_{\ell,u}$.
    - ○ Set $L_\ell = L_\ell \cup \Gamma_u$ and $\mathcal{E}_\ell = \mathcal{E}_\ell \cup \mathcal{E}_{\ell,u}$.
  - ◇ $L_\ell = L_\ell \setminus \bigcup_{i=0}^{\ell-1} L_i$.
- **Return** the edges $\bigcup_{\ell=1}^{\mathfrak{t}} \mathcal{E}_\ell$.

HTester $(\mathcal{H}, \mathfrak{Rep}, \mathcal{M}, \mathfrak{d}, \mathfrak{t})$:

- Run Random-HTraverse $(\mathcal{H}, \mathfrak{Rep}, \mathfrak{d}, \mathfrak{t})$ and let $\mathcal{E}$ be the resulted set of edges.
- If the sub-hypergraph of $\mathcal{H}$ induced by the edges $\mathcal{E}$ contains a copy of $\mathcal{M}$, then **reject**.
- If not, then **accept**.

V. Finding the first set $\mathbb{Q}_1$ of edge-disjoint colored copies of H

We now proceed with a simple construction that for a given graph G that is ε-far from H-free, finds a set $\mathbb{Q}_1$ of $\Omega_{\varepsilon,H}(|V|)$ edge-disjoint colored copies of H in G.

Lemma 20: If G is ε-far from H-free, then one can color vertices of G with $|V(H)|$ colors χ such that G has a set $\mathbb{Q}$ of at least $\frac{\varepsilon}{|E(H)|\cdot|V(H)|^{|V(H)|}} \cdot |V|$ edge-disjoint colored copies of H.

Proof. We first find the copies of H without considering the coloring of V and $V(H)$, and then we will prove the existences of the relevant coloring χ.

We find edge-disjoint copies of H in G one by one. Suppose that we have already found in G a set of k edge-disjoint copies of H, where $k < \frac{\varepsilon|V|}{|E(H)|}$. Then, since G is ε-far from H-free, the graph obtained from G by removal of the k copies of H found already (which removes $k|E(H)| < \varepsilon|V|$ edges from G) cannot be H-free, and hence G must contain a copy of H. This copy would be edge-disjoint with all copies found before, what by induction shows that G has at least $\frac{\varepsilon\cdot|V|}{|E(H)|}$ edge-disjoint copies of H.

Let $H_1, \dots, H_\ell$ be the edge-disjoint copies of H in G, with $\ell \geq \frac{\varepsilon\cdot|V|}{|E(H)|}$. Let us consider a uniformly random coloring of vertices of G (with $|V(H)|$ colors) and let X_i be the indicator random variable that H_i has all vertices of the same color as in H; let $X = \sum_{i=1}^{\ell} X_i$. Clearly, for every i, $\mathbf{Pr}[X_i = 1] = \mathbf{E}[X_i] = \frac{1}{|V(H)|^{|V(H)|}}$. Therefore, $\mathbf{E}[X] = \mathbf{E}[\sum_{i=1}^{\ell} X_i] = \sum_{i=1}^{\ell} \mathbf{E}[X_i] = \frac{\ell}{|V(H)|^{|V(H)|}}$. This implies that there is a coloring of vertices of G that has at least $\frac{\ell}{|V(H)|^{|V(H)|}} \geq \frac{\varepsilon\cdot|V|}{|E(H)|\cdot|V(H)|^{|V(H)|}}$ edge-disjoint colored copies of H. Therefore, there is a coloring χ with this property, that is, after we color vertices of G using χ, then G will have at least $\frac{\varepsilon\cdot|V|}{|E(H)|\cdot|V(H)|^{|V(H)|}}$ edge-disjoint colored copies of H that form the required set $\mathbb{Q}$. ■

Using the result from Lemma 20, from now on, we will assume that the vertices of G are colored using χ (the coloring from Lemma 20) so that G has at least $\Omega_{\varepsilon,H}(|V|)$ edge-disjoint colored copies of H.

VI. Constructing $\mathbb{Q}_{i+1}$ from $\mathbb{Q}_i$

The construction of $\mathbb{Q}_1$ from Section V is rather simple, but it is significantly more complex to define $\mathbb{Q}_2$, and then $\mathbb{Q}_3, \dots, \mathbb{Q}_{|V(H)|}$. In what follows, we will first present key intuitions in Section VI-A, then describe our framework in Sections VI-B and VI-C, and present details of the construction of $\mathbb{Q}_{i+1}$ in Section VI-D.

While our main focus is on the sets $\mathbb{Q}_i$ of edge-disjoint colored copies of H in G, in our analysis we will analyze these sets and the relevant graphs $G[\mathbb{Q}_i]$ via their suitable *hypergraph representation*. Indeed, to prove that Random-Traverse finds a copy of H, we will consider a hypergraph induced by "shrunk" copies of H defining $\mathbb{Q}_i$. The idea of this construction is two-folded:

- on one hand, using the hypergraph representation it will be easier to argue a lower bound for the probability that a copy of H is found, and
- on the other hand, the hypergraph representation will allow us to combine distinct colored copies of H (or the subgraph of H) that are undistinguishable to Random-Traverse.

A. Overview: Gadgets, hypergraph representation & their use

Our analysis relies on special structures (gadgets) in the input graph and then representing these gadgets in a succinct way using *hypergraphs*.

Let G^* be a subgraph of G that has a copy of H. Consider a subgraph H_1 of H and let $u_1, \dots, u_\ell$ be the vertices in the copy of H_1 in G^* that separate $G \setminus H_1$ from H_1, so that (cf. Figure 1):

(a) every vertex from $\{u_1, \dots, u_\ell\}$ is adjacent in G^* to some vertex $H_1 \setminus \{u_1, \dots, u_\ell\}$,

(b) every vertex in $H_1 \setminus \{u_1, \dots, u_\ell\}$ is adjacent in G^* only to vertices from H_1, and

(c) $\{u_1, \dots, u_\ell\}$ forms an independent set in H_1.

Then, we can construct a gadget to represent that copy of H_1 by removing from H_1 all vertices and edges from $H_1 \setminus \{u_1, \dots, u_\ell\}$ and replacing them by a single *hyperedge* $\{u_1, \dots, u_\ell\}$.

We will be using this construction of gadgets to model the following scenario:

- *when entering (in Random-Traverse) H_1 via any single edge incident to any vertex from the separator $u_1, \dots, u_\ell$ is sufficient to visit (with constant probability) all edges in H_1.*

Therefore, for the analysis, this will correspond to the situation that

- there is a hyperedge $\{u_1, \dots, u_\ell\}$, and by visiting this hyperedge (in the hypergraph), the algorithm will visit (with constant probability) all edges in H_1 (in the original graph), and will be able to continue the search from *all* separating vertices $u_1, \dots, u_\ell$.

Furthermore, the gadgets can be also helpful in the analysis of "substitutable" copies of a subgraph of H. Suppose that for a subgraph H_1 of H, the separator (as defined above) is identical in multiple copies, that is, vertices $u_1, \dots, u_\ell$ form the separator in multiple edge-disjoint copies of H_1. Then, we have multiple hyperedges $\{u_1, \dots, u_\ell\}$ and their multiplicity represents the fact that *to find a copy of H_1 it is enough to visit just one of the hyperedges* $\{u_1, \dots, u_\ell\}$. In particular, if u_1 is incident to multiple copies of the identical hyperedge $\{u_1, \dots, u_\ell\}$, then the probability that the process will visit H_1 starting from u_1 increases with this multiplicity. And so, if the multiplicity is of order $\deg_G(u_1)$, then after reaching vertex u_1, the Random-HTraverse algorithm (cf. Section IV-A3) will visit the entire H_1 with a constant probability.

The central idea behind the gadgets as described above is to use them repeatedly to transform a subgraph of G into a sub-hypergraph representing a smaller subgraph of G for which we can easily analyze the Random-HTraverse algorithm.

B. Shrinking H and hypergraph representation of H by $\mathcal{M}_i$

We will begin with an iterative procedure that gradually shrinks H into a single vertex. This procedure processes H and its contractions in a form of a *hypergraph*. (See also Figures 2–4.)

Let us consider an arbitrary numbering of the vertices of H, $v_1, v_2, \dots, v_{|V(H)|}$; this order is not known in advance and is independent of the coloring of H (in fact, the order will be determined by the structure of G, and finding the right order $v_1, v_2, \dots, v_{|V(H)|}$ is the central part of our analysis in the next sections, finalized in Lemma 33). In our analysis, we will perform a sequence of transformations on H, each transformation converting some hypergraph $\mathcal{M}_i$ corresponding to H into some other hypergraph $\mathcal{M}_{i+1}$ corresponding to H, $1 \le i \le |V(H)| - 1$ (cf. Figures 2–4), such that:

- $\mathcal{M}_1 := H$, and
- $\mathcal{M}_{i+1}$ is obtained from $\mathcal{M}_i$ by contracting vertex v_i to its neighbors as follows:
  - ◇ let $\mathcal{N}_i$ be the set of all neighbors of v_i in $\mathcal{M}_i$; contract v_i to its neighbors by removing v_i from $\mathcal{M}_i$ and then adding a new hyperedge consisting of vertices in $\mathcal{N}_i$.

We will want to maintain information about all vertices which have been contracted to create a given hyperedge (e.g., in Figure 1, these would be vertices $\{1, 5\}$) and so we will *label* the hyperedges. We will denote the label of an edge $\mathfrak{e}$ by $\sigma(\mathfrak{e})$. A regular edge e (original edge from $E(H)$) has an empty label, i.e., $\sigma(e) = \emptyset$, and if $\mathcal{E}_i$ denotes the set of edges/hyperedges incident to vertex v_i in $\mathcal{M}_i$, then the new hyperedge $\mathcal{N}_i$ obtained by contraction of v_i will have label $\sigma(\mathcal{N}_i) = \{v_i\} \cup \bigcup_{\mathfrak{e} \in \mathcal{E}_i} \sigma(\mathfrak{e})$ (i.e., its label is the union of $\{v_i\}$ and the union of the labels of the edges in $\mathcal{E}_i$).

Furthermore, we will also have *colored label* σ^* of any edge $\mathfrak{e}$, defined as the set of the colors of the vertices defining the

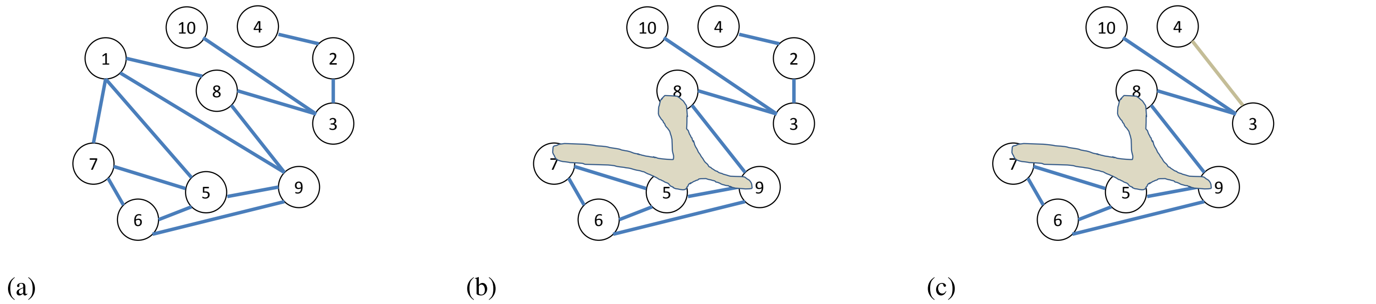


Fig. 2. Consider the input graph G in Figure (a) and consider the process of shrinking G, as described in Section VI-B. (b) presents contraction of vertex 1 and adding of hyperedge $\{5, 7, 8, 9\}$ (with label $\{1\}$). (c) After contracting vertex 2 and adding hyperedge $\{3, 4\}$ (with label $\{2\}$).

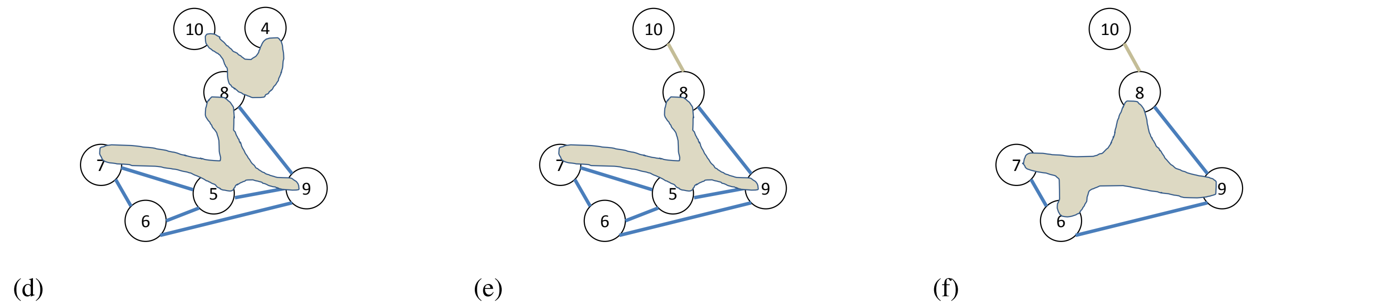


Fig. 3. Continuing the example from Figure 2: (d) After contracting vertex 3 and adding hyperedge $\{4, 8, 10\}$ (with label $\{2, 3\}$). (e) After contracting vertex 4 and adding hyperedge $\{8, 10\}$ (with label $\{2, 3, 4\}$). (f) After contracting vertex 5 and adding hyperedge $\{6, 7, 8, 9\}$ (with label $\{1, 5\}$).

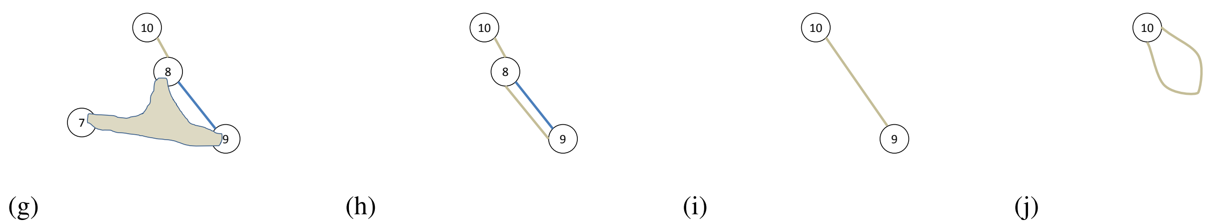


Fig. 4. Continuing the example from Figures 2 and 3: (g) After contracting vertex 6 and adding hyperedge $\{7, 8, 9\}$ (with label $\{1, 5, 6\}$). (h) After contracting vertex 7 and adding hyperedge $\{8, 9\}$ (with label $\{1, 5, 6, 7\}$; note that as the result, we have two parallel edges between 8 and 9, but each of these edges is different, one corresponds to a direct edge between 8 and 9 with label $\emptyset$, and another corresponds to the gadget with separator $\{8, 9\}$ and internal vertices $\{1, 5, 6, 7\}$ (as shown by the label)). (i) After contracting vertex 8 and adding hyperedge $\{9, 10\}$ (with label $\{1, 2, 3, 4, 5, 6, 7, 8\}$). (j) After contracting vertex 9 and hyperedge $\{10\}$ (with label $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$).

label of $\mathfrak{e}$, that is, $\sigma^*(\mathfrak{e}) = \{\chi(u) : u \in \sigma(\mathfrak{e})\}$. (Note that if $\sigma(\mathfrak{e}) = \emptyset$ then $\sigma^*(\mathfrak{e}) = \emptyset$.)

We will also use the following notion.

Definition 21: If in our construction, in $\mathcal{M}_i$, we had edges $\mathfrak{e}_1, \dots, \mathfrak{e}_\ell$ incident to v_i, then we will say that the newly created *hyperedge $\mathcal{N}_i$ in $\mathcal{M}_{i+1}$ is modeled by edges $\mathfrak{e}_1, \dots, \mathfrak{e}_\ell$ in $\mathcal{M}_i$.*

Let us note that the construction above allows "*selfloops*," that is, hyperedges consisting of a single vertex, and that it allows multiple copies of hyperedges on the same vertex set (see, e.g., Figure 4 (h) or Figure 5, and one could have many copies of hyperedges even with more than two vertices). An important feature of the latter case is that all these hyperedges will be considered as different hyperedges, since they correspond to different subgraphs of H and have different labels. Note also that all labels are disjoint (i.e., $\sigma(\mathfrak{e}_1) \cap \sigma(\mathfrak{e}_2) = \emptyset$ for any distinct hyperedges $\mathfrak{e}_1, \mathfrak{e}_2$ in $\mathcal{M}_i$).

C. Shrinking copies of H in G

The central idea of our analysis is to mimic the corresponding transformation of H (as described in Section VI-B) in all relevant copies of H in sets $\mathbb{Q}_1, \mathbb{Q}_2, \dots$, and then, instead of searching for a copy of H in $G[\mathbb{Q}_1], G[\mathbb{Q}_2], \dots$, to search for copies of $\mathcal{M}_1, \mathcal{M}_2, \dots$ in the corresponding shrunk hypergraphs $\mathcal{H}_1(\mathbb{Q}_1), \mathcal{H}_2(\mathbb{Q}_2), \dots$. Then, we will argue that finding a copy of H in G is (almost) as easy as finding a copy of $\mathcal{M}_1$ in $\mathcal{H}_1(\mathbb{Q}_1)$, which in turn can be reduced (by paying a small

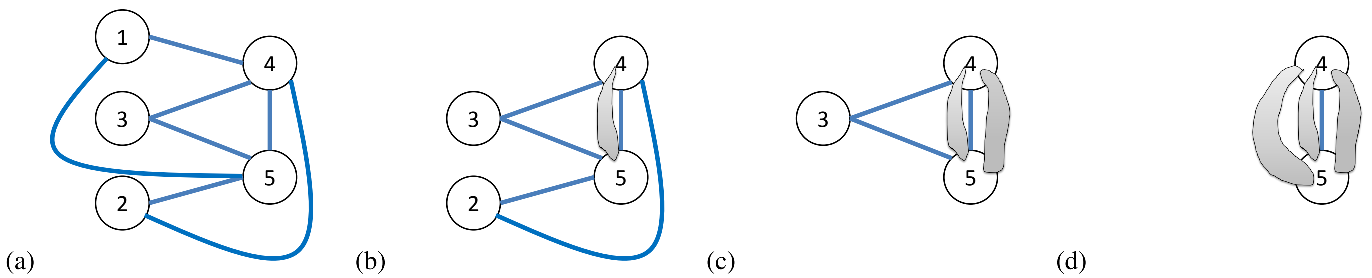


Fig. 5. Construction of hypergraphs (a) $\mathcal{M}_1$, (b) $\mathcal{M}_2$, (c) $\mathcal{M}_3$, (d) $\mathcal{M}_4$, with multiple hyperedges $\{4,5\}$.

price) to finding a copy of $\mathcal{M}_2$ in $\mathcal{H}_2(\mathbb{Q}_2)$, and so on, reducing everything to finding a copy of $\mathcal{M}_{|V(H)|}$ in $\mathcal{H}_{|V(H)|}(\mathbb{Q}_{|V(H)|})$. And then, since $\mathcal{M}_{|V(H)|}$ has only a single vertex, we would hope that finding its copy in $\mathcal{H}_{|V(H)|}(\mathbb{Q}_{|V(H)|})$ is easy.

In order to incorporate this approach, we will transform appropriate subgraphs of G into a sequence of hypergraphs, such that after i transformations, every relevant copy of H is shrunk into $\mathcal{M}_{i+1}$. (Let us emphasize that this step relies on the choice of vertex v_i — which is the same in all copies of H — to be determined by the structure of $\mathbb{Q}_i$, as described in Lemma 33.) In particular, we will mimic the corresponding transformation on $\mathbb{Q}_1, \mathbb{Q}_2, \dots$ as follows.

We consider a hypergraph, denoted by $\mathcal{H}_i(\mathbb{Q}_i)$, corresponding to $\mathbb{Q}_i$, which has

- vertex set $V(\mathcal{H}_i(\mathbb{Q}_i)) = V \setminus \{u \in V : \chi(u) \in \{\chi(v_j) : j < i\}\}$ (vertices[4] in G that have colors of vertices $\{v_i, \dots, v_{|V(H)|}\}$, that is, that have not been contracted in $\mathcal{M}_i$ yet), and
- edge set formed by an edge-disjoint collection of copies of $\mathcal{M}_i$ (we allow hyperedges to have some multiplicity).

Then, for some carefully chosen set $\mathbb{Q}_{i+1} \subseteq \mathbb{Q}_i$, a new hypergraph $\mathcal{H}_{i+1}(\mathbb{Q}_{i+1})$ is obtained from $\mathcal{H}_i(\mathbb{Q}_i)$ by

- removing all hyperedgescorresponding to the edge-disjoint copies of H in $\mathbb{Q}_i \setminus \mathbb{Q}_{i+1}$ and
- then taking the set $\mathbb{Q}_{i+1}$ of copies of H and shrinking them, in the same way as $\mathcal{M}_i$ is transformed into $\mathcal{M}_{i+1}$:

- Select a vertex $v_i \in V(H)$.
- Simultaneously, contract every vertex $u \in V(\mathcal{H}_i(\mathbb{Q}_i))$ with $\chi(u) = \chi(v_i)$ as follows:
  - ◇ for every colored copy $\mathfrak{h}$ of H in $\mathbb{Q}_{i+1}$ that contains vertex u:
    - ∘ add a new hyperedge consisting of vertices in $\mathcal{N}_i^{\mathfrak{h}}\langle u\rangle$, where $\mathcal{N}_i^{\mathfrak{h}}\langle u\rangle$ is the set of neighbors of u in $\mathfrak{h}$ (in the hypergraph $\mathcal{H}_i(\mathbb{Q}_i)$) other than u (that is, $u \notin \mathcal{N}_i^{\mathfrak{h}}\langle u\rangle$);
  - ◇ remove vertex u (with all incident edges from $\mathcal{H}_i(\mathbb{Q}_i)$).

Notice that in our construction of $\mathcal{H}_{i+1}(\mathbb{Q}_{i+1})$ we are removing all vertices $u \in V$ with color $\chi(u) = \chi(v_i)$. And so, in particular, $V(\mathcal{H}_{i+1}(\mathbb{Q}_{i+1})) = V \setminus \{u \in V : \chi(u) \in \{\chi(v_j) : j \le i\}\}$.

Furthermore, since we contract only vertices of color $\chi(v_i)$ and since these vertices are independent in $G[\mathbb{Q}_{i+1}]$ (indeed, since $\mathbb{Q}_{i+1}$ is the set of edge-disjoint colored copies of H, $G[\mathbb{Q}_{i+1}]$ does not have monochromatic edges), the operation above is well defined and the contractions of all vertices of color $\chi(v_i)$ can be performed independently in all copies of H in $\mathbb{Q}_{i+1}$. This yields an equivalent definition:

Remark 22: The following is an equivalent definition of $\mathcal{H}_{i+1}(\mathbb{Q}_{i+1})$:

- Start with graph $G[\mathbb{Q}_{i+1}]$.
- For every copy $\mathfrak{h}$ of H in $\mathbb{Q}_{i+1}$, perform the shrinking of H into hypergraph $\mathcal{M}_{i+1}$.
- Combine all copies of $\mathcal{M}_{i+1}$ obtained in that way.
- Remove all vertices u with $\chi(u) = \chi(v_j)$ for $j \le i$ that do not belong to any copy of $\mathbb{Q}_{i+1}$.

The fact that this description is correct follows from the fact that the shrinking of different copies of H can be performed independently because of vertex coloring, which ensures that if we contract a vertex u with $\chi(u) = \chi(v_j)$ and create a new edge $\mathcal{N}_j^{\mathfrak{h}}\langle u\rangle$, then this construction can be performed independently for different copies of H.

Notice that because of the construction above, to define $\mathcal{H}_{i+1}(\mathbb{Q}_{i+1})$, we do not need to consider the constructions of $\mathcal{H}_1(\mathbb{Q}_1), \mathcal{H}_2(\mathbb{Q}_2), \dots, \mathcal{H}_i(\mathbb{Q}_i)$ one after another, but we could do it with the constructions of $\mathcal{H}_1(\mathbb{Q}_{i+1}), \mathcal{H}_2(\mathbb{Q}_{i+1}), \dots, \mathcal{H}_i(\mathbb{Q}_{i+1})$, and from $\mathcal{H}_i(\mathbb{Q}_{i+1})$ to build $\mathcal{H}_{i+1}(\mathbb{Q}_{i+1})$.

(Note that the vertex set of $\mathcal{H}_{i+1}(\mathbb{Q}_{i+1})$ is $V(\mathcal{H}_{i+1}(\mathbb{Q}_{i+1})) = V \setminus \{u \in V : \chi(u) \in \{\chi(v_j) : j \le i\}\}$. Further, observe that $\mathcal{H}_{i+1}(\mathbb{Q}_{i+1})$ may have (isolated) vertices u that do not belong to any copy of $\mathbb{Q}_{i+1}$.)

The construction above maintains a relationship between edges in $\mathcal{H}_i(\mathbb{Q}_i)$ and edges in $\mathcal{M}_i$.

Definition 23: **(Corresponding edges)** If $\mathfrak{e}$ is an edge in $\mathcal{H}_i(\mathbb{Q}_i)$ then the *corresponding edge to* $\mathfrak{e}$ *in* $\mathcal{M}_i$ is edge $\mathfrak{e}'$ in $\mathcal{M}_i$ such that the colors of vertices in $\mathfrak{e}$ are the same as the colors of vertices in $\mathfrak{e}'$ (i.e., $\{\chi(x) : x \in \mathfrak{e}\} = \{\chi(v_j) : v_j \in \mathfrak{e}'\}$), and the colored labels of $\mathfrak{e}$ and $\mathfrak{e}'$ are the same too (i.e., $\sigma^*(\mathfrak{e}) = \sigma^*(\mathfrak{e}^*)$).

Notice that every edge in $\mathcal{H}_i(\mathbb{Q}_i)$ has a unique corresponding edge in $\mathcal{M}_i$. Furthermore, for any edge $\mathfrak{e}'$ in $\mathcal{M}_i$, the

[4]Let us first remind that we are assuming that the vertices of G are colored using χ so that G has at least $\Omega_{\varepsilon,H}(|V|)$ edge-disjoint colored copies of H, as promised by Lemma 20.

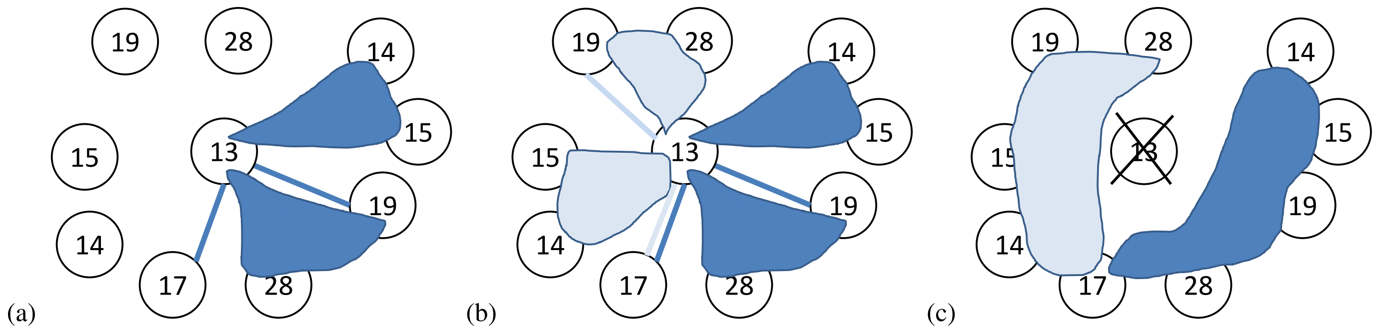


Fig. 6. Operation of contracting vertex of color 13 (numbers depicted correspond here to the colors) to define new hypergraph $\mathcal{H}_{14}(\mathbb{Q}_{14})$. (a) Describes the edges of a *single copy* of $\mathcal{M}_{13}$ incident to vertex of color 13 (in the center) in $\mathcal{H}_{13}(\mathbb{Q}_{13})$. (b) Describes the edges of *two copies* of $\mathcal{M}_{13}$ incident to vertex of color 13 in $\mathcal{H}_{13}(\mathbb{Q}_{13})$. (Notice that in this case, vertex of color 13 is not safe.) (c) Describes the situation of contracting vertex of color 13, which creates two new hyperedges, and removal of vertex of color 13 and all incident edges. This defines $\mathcal{H}_{14}(\mathbb{Q}_{14})$.

number of edges in $\mathcal{H}_i(\mathbb{Q}_i)$ corresponding to edge $\mathfrak{e}'$ in $\mathcal{M}_i$ is exactly $|\mathbb{Q}_i|$.

Next, we can also mimic Definition 21 in the context of our construction here as follows:

Definition 24: **(Modeling edges in $\mathcal{H}_{i+1}(\mathbb{Q}_{i+1})$ by edges in $\mathcal{H}_i(\mathbb{Q}_i)$)** Let u be a vertex in $\mathcal{H}_i(\mathbb{Q}_i)$ with $\chi(u) = \chi(v_i)$. Let $\mathfrak{h}$ be a colored copy of H in $\mathbb{Q}_{i+1}$ that contains vertex u. Let $\mathfrak{e}_1, \dots, e_\ell$ be the edges incident to u $\mathcal{H}_i(\mathbb{Q}_i)$ corresponding to the copy $\mathfrak{h}$. Then, we will say that the newly created hyperedge $\mathcal{N}_i^{\mathfrak{h}}\langle u\rangle$ in $\mathcal{H}_{i+1}(\mathbb{Q}_{i+1})$ is *modeled by edges* $\mathfrak{e}_1, \dots, \mathfrak{e}_\ell$ in $\mathcal{H}_i(\mathbb{Q}_i)$.

Now, we are ready to formalize the process of finding a colored copy of $\mathcal{M}_i$ in a hypergraph.

Definition 25: **(Finding a colored copy of $\mathcal{M}_i$)** Let $v_i, \dots, v_{|V(H)|}$ be the vertices in $\mathcal{M}_i$. We say that HT-ester $(\mathcal{H}, \mathfrak{Rep}, \mathcal{M}_i, \mathfrak{d}, \mathfrak{t})$ finds a colored copy of $\mathcal{M}_i$ in $\mathcal{H}$ if the corresponding algorithm Random-HTraverse $(\mathcal{H}, \mathfrak{Rep}, \mathfrak{d}, \mathfrak{t})$ returns a set of edges $\mathcal{E}$, such that

- the sub-hypergraph of $\mathcal{H}$ induced by the edges $\mathcal{E}$ contains vertices $x_i, \dots, x_{|V(H)|}$ such that for every edge/hyperedge $\mathfrak{e}$ in $\mathcal{M}_i$, $\mathcal{E}$ contains an edge corresponding to $\mathfrak{e}$, or equivalently,
  - ⋄ $\chi(x_j) = \chi(v_j)$ for every j, $i \le j \le |V(H)|$, and
  - ⋄ for every edge $\{v_{j_1}, \dots, v_{j_r}\}$ in $\mathcal{M}_i$, $\mathcal{E}$ contains edge $\{x_{j_1}, \dots, x_{j_r}\}$.

1) Adjusting for planar graphs: safe vertices and consistent hypergraphs: In our construction we will require more properties from the contractions defining $\mathcal{H}_{i+1}(\mathbb{Q}_{i+1})$. To maintain *some basic properties of planar graphs* (which are required by our analysis), we will want to model the operation of contraction of a vertex u as the standard vertex contraction of u to one of its neighbors. For that, we will need an additional, *stronger property*:

- we want to ensure that all contractions in $\mathcal{H}_i(\mathbb{Q}_i)$ corresponding to the contraction of v_i in $\mathcal{M}_i$ are *consistent*, that is, the contraction of u is the same in every colored copy of H that contains u (that is, for every vertex u in with $\chi(u) = \chi(v_i)$, for any two colored copies $\mathfrak{h}_1, \mathfrak{h}_2$ of H in $\mathbb{Q}_{i+1}$ containing vertex u, we have $\mathcal{N}_i^{\mathfrak{h}_1}\langle u\rangle = \mathcal{N}_i^{\mathfrak{h}_2}\langle u\rangle$).

To facilitate this property, we will use the following definitions.

Definition 26: **(Safe vertices)** Let $\mathbb{Q}_i$ be a set of edge-disjoint colored copies of H in G and let $\mathbb{Q} \subseteq \mathbb{Q}_i$. We call a vertex $u \in V(\mathcal{H}_i(\mathbb{Q}_i))$ *safe* (with respect to $\mathbb{Q}$ and $\mathcal{H}_i(\mathbb{Q}_i)$) if for all colored copies $\mathfrak{h} \in \mathbb{Q}$ of H that contain u, the sets $\mathcal{N}_i^{\mathfrak{h}}\langle u\rangle$ are the same.

Remark 27: Note that Definition 26 means that for every safe vertex u with respect to $\mathbb{Q}$ and $\mathcal{H}_i(\mathbb{Q}_i)$, not only all edges incident to u correspond to the edges from $\mathcal{M}_i$ incident to vertex v in $\mathcal{M}_i$ with $\chi(u) = \chi(v)$, but also, if u is incident to r edges in $\mathcal{H}_i(\mathbb{Q}_i)$ and v is incident to edges $\mathfrak{e}_1, \dots, \mathfrak{e}_\ell$ in $\mathcal{M}_i$, then

(i) we can partition the edges incident to u into ℓ groups, each group corresponding to one of the edges $\mathfrak{e}_1, \dots, \mathfrak{e}_\ell$ in $\mathcal{M}_i$, each group of the same size r/ℓ, such that two edges $\mathfrak{e}', \mathfrak{e}''$ from the same group have the same colored label (i.e., $\sigma^*(\mathfrak{e}') = \sigma^*(\mathfrak{e}'')$) and are defined by the same vertices (i.e., for every vertex x, $x \in \mathfrak{e}'$ iff $x \in \mathfrak{e}''$);

(ii) $|\mathcal{N}_i^{\mathfrak{h}}\langle u\rangle| = |\bigcup_{j=1}^{\ell} \mathfrak{e}_j \setminus \{v\}|$, that is, u has as many neighbors in $\mathcal{H}_i(\mathbb{Q}_i)$ as v has in $\mathcal{M}_i$;

(iii) $\{\chi(x) : x \in \mathcal{N}_i^{\mathfrak{h}}\langle u\rangle\} = \{\chi(x) : x \in \bigcup_{j=1}^{\ell} \mathfrak{e}_j \setminus \{v\}\}$.

Our next iterative definition extends the notion of safe vertices to the entire hypergraph.

Definition 28: **(Consistent hypergraphs)** For any set $\mathbb{Q}_1$ of edge-disjoint colored copies of H in G, the hypergraph $\mathcal{H}_1(\mathbb{Q}_1)$ (which is equal to the graph $G[\mathbb{Q}_1]$) is called *consistent* (for $\mathbb{Q}_1$).

Let $\mathbb{Q}_i$ be a set of edge-disjoint colored copies of H in G and let $\mathbb{Q}_{i+1} \subseteq \mathbb{Q}_i$. If hypergraph $\mathcal{H}_i(\mathbb{Q}_i)$ is consistent for $\mathbb{Q}_i$, then hypergraph $\mathcal{H}_{i+1}(\mathbb{Q}_{i+1})$ obtained from $\mathcal{H}_i(\mathbb{Q}_i)$ is called *consistent* (for $\mathbb{Q}_{i+1}$) if every vertex $u \in V(\mathcal{H}_i(\mathbb{Q}_i))$ with $\chi(u) = \chi(v_i)$ is safe with respect to $\mathbb{Q}_{i+1}$ and $\mathcal{H}_i(\mathbb{Q}_i)$.

2) Central property of consistent hypergraphs via shadow graphs: With the notion of safe vertices and consistent hypergraphs, we can now present the following central lemma that shows that the neighborhood of vertices in consistent

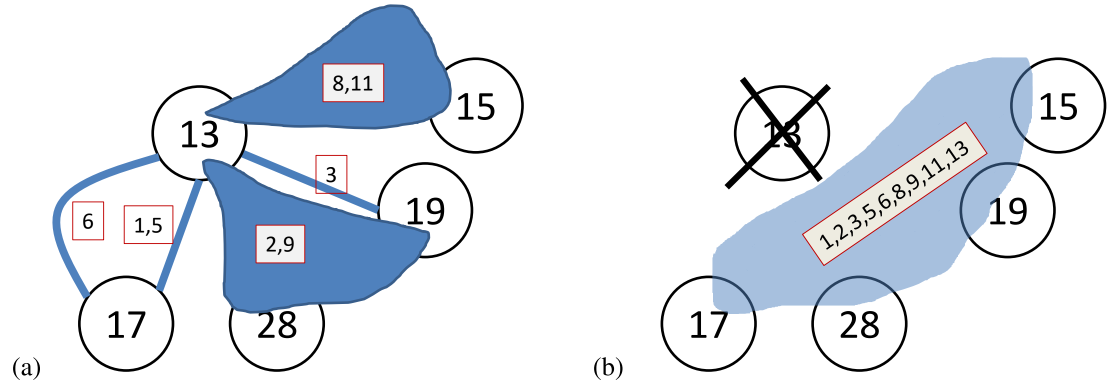


Fig. 7. Operation of contracting safe vertex of color 13 (numbers depicted in the vertices correspond to the colors, and number depicted next to the edges correspond to the colored labels of the edges) to define new hypergraph $\mathcal{H}_{14}(\mathbb{Q}_{14})$. (a) Describes the edges of a single copy of $\mathcal{M}_{13}$ incident to vertex of color 13 (in the center) in $\mathcal{H}_{13}(\mathbb{Q}_{13})$. Since we want vertex of color 13 to be safe, it is possible there are many more copies of $\mathcal{M}_{13}$ incident to that vertex, in which case all of them use identical edges as in the depicted single copy (here identical means: on the same vertex set, with the same colored labels, but with distinct labels). (b) Describes the situation after contracting vertex of color 13 in $\mathcal{H}_{14}(\mathbb{Q}_{14})$. Notice that the new edge in $\mathcal{H}_{14}(\mathbb{Q}_{14})$ has colored label $\{1, 2, 3, 5, 6, 8, 9, 11, 13\}$ and is modeled by five edges in $\mathcal{H}_{13}(\mathbb{Q}_{13})$.

hypergraphs can be modeled by some semi-planar structures, which we will call *shadow graphs*, that are a union of at most $|V(H)|$ simple planar graphs.

Lemma 29: Let $\mathbb{Q}_i$ be a set of edge-disjoint colored copies of H in G and let $\mathcal{H}_i(\mathbb{Q}_i)$ be a hypergraph consistent for $\mathbb{Q}_i$. Then, there is a simple graph $\mathfrak{G}(\mathcal{H}_i(\mathbb{Q}_i))$,

(a) with the vertex set equal to the set of all non-isolated vertices in $\mathcal{H}_i(\mathbb{Q}_i)$,

(b) that is a union of at most $|V(H)|$ simple planar graphs, and

(c) such that for any distinct $x, y \in V(\mathcal{H}_i(\mathbb{Q}_i))$, x is adjacent to y in $\mathcal{H}_i(\mathbb{Q}_i)$ if and only if x is adjacent to y in $\mathfrak{G}(\mathcal{H}_i(\mathbb{Q}_i))$.

The simple graph $\mathfrak{G}(\mathcal{H}_i(\mathbb{Q}_i))$ in Lemma 29 will be called the *shadow graph* of $\mathcal{H}_i(\mathbb{Q}_i)$.

We consider the characterization provided in Lemma 29 to be one of the most interesting and highly non-trivial contributions of this paper. This is the key tool that allows us to facilitate the approach presented in the paper. (The proof of Lemma 29 is deferred to the full version of the paper [13].)

3) Finding many safe vertices of the same color: The main use of Lemma 29 is to show that even though the use of the hypergraphs $\mathcal{H}_1(\mathbb{Q}_1), \mathcal{H}_2(\mathbb{Q}_2), \dots$ looses some basic properties of planar graphs, our use of consistent hypergraphs allows us to apply Lemma 29 to maintain some weaker, but still similar properties of the hypergraphs $\mathcal{H}_1(\mathbb{Q}_1), \mathcal{H}_2(\mathbb{Q}_2), \dots$. We begin with the following lemma that shows that the hypergraphs will have a constant fraction of vertices of low degrees. The proof of our next Lemma 30 extends the approach used earlier in the context of planar graphs from [11]; we defer the proof to the full version of the paper [13].

Lemma 30: Let $\mathbb{Q}_i$ be a set of edge-disjoint colored copies of H in G and let $\mathcal{H}_i(\mathbb{Q}_i)$ be a hypergraph consistent for $\mathbb{Q}_i$. Then, there is a set $\mathbb{Q} \subseteq \mathbb{Q}_i$ of size at least $\frac{|\mathbb{Q}_i|}{4|V(H)|+2}$ such that in the hypergraph $\mathcal{H}_i(\mathbb{Q})$, every copy of H in $\mathbb{Q}$ has a vertex with at most $6|V(H)|$ distinct neighbors.

Our next lemma follows the arguments used in a related proof from [11] and shows that if there is a color with all vertices having a small number of neighbors in $\mathcal{H}_i(\mathbb{Q})$ for $\mathbb{Q} \subseteq \mathbb{Q}_i$, then we can always find a large subset of $\mathbb{Q}$ with all vertices of that color being safe.

Lemma 31: Let $\mathbb{Q}_i$ be a set of edge-disjoint colored copies of H in G such that $\mathcal{H}_i(\mathbb{Q}_i)$ is a hypergraph consistent for $\mathbb{Q}_i$. Let $\mathfrak{c}$ be a color of a vertex in $\{1, \dots, |V(H)|\} \setminus \{\chi(v_j) : j < i\}$. Let $\mathbb{Q} \subseteq \mathbb{Q}_i$ such that every colored copy of H in $\mathbb{Q}$ has vertex of color $\mathfrak{c}$ with at most $6|V(H)|$ distinct neighbors in $\mathcal{H}_i(\mathbb{Q})$. Then there is a subset $\mathbb{Q}' \subseteq \mathbb{Q}$, $|\mathbb{Q}'| \geq \frac{|\mathbb{Q}|}{(6|V(H)|)^{|V(H)|}}$, such that every colored copy $\mathfrak{h}$ of H in $\mathbb{Q}'$ has vertex of color $\mathfrak{c}$ safe with respect to $\mathbb{Q}'$ and $\mathcal{H}_i(\mathbb{Q}_i)$.

Proof. Let $\mathfrak{c}_1, \dots, \mathfrak{c}_\ell$ be the colors of vertices adjacent to vertex of color $\mathfrak{c}$ in $\mathcal{M}_i$ (notice that $\mathfrak{c}$ may be among these colors). For each non-isolated vertex u in $\mathcal{H}_i(\mathbb{Q})$ of color $\mathfrak{c}$, for every color $\mathfrak{c}_s$, $1 \leq s \leq \ell$, select i.u.r. one of its neighbors $u_{\langle s \rangle}$ in $\mathcal{H}_i(\mathbb{Q})$ of color $\mathfrak{c}_s$. Next, remove from $\mathbb{Q}$ every copy of $\mathfrak{h}$ of H in $\mathcal{H}_i(\mathbb{Q})$ containing vertex u unless the vertices from this copy incident to u are the selected ℓ neighbors $u_{\langle 1 \rangle}, u_{\langle 2 \rangle}, \dots, u_{\langle \ell \rangle}$. Let $\mathbb{Q}'$ be the set of remaining copies of H in $\mathcal{H}_i(\mathbb{Q})$.

Our construction ensures that every remaining non-isolated vertex u of color $\mathfrak{c}$ is safe with respect to $\mathbb{Q}'$ and $\mathcal{H}_i(\mathbb{Q}_i)$. Furthermore, since every vertex of color $\mathfrak{c}$ has at most $6|V(H)|$ distinct neighbors (taking into account self-loops) in $\mathcal{H}_i(\mathbb{Q})$, the probability that a fixed copy of $\mathfrak{h}$ in $\mathbb{Q}$ is not deleted by the process above is at least $(6|V(H)|)^{-\ell}$. Therefore the expected size of $\mathbb{Q}'$ is at least $(6|V(H)|)^{-\ell} \cdot |\mathbb{Q}|$, and therefore, there exists a set $\mathbb{Q}'$ of that size that satisfies the lemma. ■

With Lemmas 30 and 31 at hand, we are now ready to present the main result of this section.

Lemma 32: Let $\mathbb{Q}_i$ be a set of edge-disjoint colored copies of H in G and let $\mathcal{H}_i(\mathbb{Q}_i)$ be a hypergraph consistent for $\mathbb{Q}_i$. Then, there is color $\mathfrak{c}$ in $\{1, \dots, |V(H)|\} \setminus \{\chi(v_j) : j < i\}$ and a set $\mathbb{Q}^* \subseteq \mathbb{Q}_i$ of size at least $\frac{|\mathbb{Q}_i|}{(6|V(H)|)^{|V(H)|+2}}$ such that every colored copy $\mathfrak{h}$ of H in $\mathbb{Q}^*$ has vertex of color $\mathfrak{c}$ safe with respect to $\mathbb{Q}^*$ and $\mathcal{H}_i(\mathbb{Q}_i)$.

Proof. By Lemma 30, there is a set $\widehat{\mathbb{Q}} \subseteq \mathbb{Q}_i$, $|\widehat{\mathbb{Q}}| \geq \frac{|\mathbb{Q}_i|}{4|V(H)|+2}$, such that every colored copy of H in $\widehat{\mathbb{Q}}$ has a vertex with at most $6|V(H)|$ distinct neighbors in $\mathcal{H}_i(\widehat{\mathbb{Q}})$. For a color $\mathfrak{c}^* \in \{1, \dots, |V(H)|\} \setminus \{\chi(v_j) : j < i\}$, let $\widehat{\mathbb{Q}}_{\mathfrak{c}^*}$ be the subset of $\widehat{\mathbb{Q}}$ such that every copy of H in $\widehat{\mathbb{Q}}_{\mathfrak{c}^*}$ has a vertex of color $\mathfrak{c}^*$ with at most $6|V(H)|$ distinct neighbors in the hypergraph $\mathcal{H}_i(\widehat{\mathbb{Q}})$. Since $\bigcup_{\mathfrak{c}^*} \widehat{\mathbb{Q}}_{\mathfrak{c}^*} = \widehat{\mathbb{Q}}$, there is one color $\mathfrak{c} \in \{1, \dots, |V(H)|\} \setminus \{\chi(v_j) : j < i\}$ such that $|\widehat{\mathbb{Q}}_{\mathfrak{c}}| \geq \frac{1}{|V(H)|} \cdot |\widehat{\mathbb{Q}}| \geq \frac{|\mathbb{Q}_i|}{(4|V(H)|+2) \cdot |V(H)|} \geq \frac{|\mathbb{Q}_i|}{(6|V(H)|)^2}$ and every copy of H in $\widehat{\mathbb{Q}}_{\mathfrak{c}}$ has a vertex of color $\mathfrak{c}$ with at most $6|V(H)|$ distinct neighbors in $\mathcal{H}_i(\widehat{\mathbb{Q}})$, and hence also in $\mathcal{H}_i(\widehat{\mathbb{Q}}_{\mathfrak{c}})$. Therefore, we can take such set $\widehat{\mathbb{Q}}_{\mathfrak{c}}$ as set $\mathbb{Q}$ in Lemma 31, to conclude that there is a subset $\mathbb{Q}' \subseteq \widehat{\mathbb{Q}}_{\mathfrak{c}}$, $|\mathbb{Q}'| \geq \frac{|\widehat{\mathbb{Q}}_{\mathfrak{c}}|}{(6|V(H)|)^{|V(H)|}} \geq \frac{|\mathbb{Q}_i|}{(6|V(H)|)^{|V(H)|+2}}$, such that every colored copy $\mathfrak{h}$ of H in $\mathbb{Q}'$ has vertex of color $\mathfrak{c}$ safe with respect to $\mathbb{Q}'$ and $\mathcal{H}_i(\mathbb{Q}_i)$. ■

D. Constructing set $\mathbb{Q}_{i+1}$ of edge-disjoint colored copies of H and $\mathcal{H}_{i+1}(\mathbb{Q}_{i+1})$

Now we are ready to define our construction of the set $\mathbb{Q}_{i+1}$ of edge-disjoint colored copies of H obtained as a subgraph of $\mathbb{Q}_i$, and with this, to define the hypergraph $\mathcal{H}_{i+1}(\mathbb{Q}_{i+1})$ from $\mathcal{H}_i(\mathbb{Q}_i)$.

Let $\mathbb{Q}_i$ be a set of edge-disjoint colored copies of H in G, where $\mathcal{H}_i(\mathbb{Q}_i)$ is a hypergraph consistent for $\mathbb{Q}_i$. We apply Lemma 32 to choose color $\mathfrak{c}$ in $\{1, \dots, |V(H)|\} \setminus \{\chi(v_j) : j < i\}$ and a set $\mathbb{Q}^* \subseteq \mathbb{Q}_i$ of size at least $\frac{|\mathbb{Q}_i|}{(6|V(H)|)^{|V(H)|+2}}$ such that every colored copy $\mathfrak{h}$ of H in $\mathbb{Q}^*$ has vertex of color $\mathfrak{c}$ safe with respect to $\mathbb{Q}^*$ and $\mathcal{H}_i(\mathbb{Q}_i)$ (that is, for every vertex u with $\chi(u) = \mathfrak{c}$, all colored copies $\mathfrak{h} \in \mathbb{Q}_{i+1}$ of H that contain u have identical sets $\mathcal{N}_i^{\mathfrak{h}}\langle u \rangle$ in $\mathcal{H}_i(\mathbb{Q}_i)$). Then, we define $\mathbb{Q}_{i+1} := \mathbb{Q}^*$ and select vertex v_i to be the vertex of color $\mathfrak{c}$ in H.

With so defined vertex v_i, we can immediately construct the hypergraph $\mathcal{H}_{i+1}(\mathbb{Q}_{i+1})$ (from the hypergraph $\mathcal{H}_i(\mathbb{Q}_i)$). The details of the construction have been presented in Section VI-C, and it required the choice of set $\mathbb{Q}_{i+1}$ and of vertex v_i among the vertices in $V(H) \setminus \{v_1, \dots, v_{i-1}\}$.

By Lemma 32 (cf. Definition 28 of consistent hypergraphs), this immediately gives the following lemma.

Lemma 33: Let $\mathbb{Q}_i$ be a set of edge-disjoint colored copies of H in G and let $\mathcal{H}_i(\mathbb{Q}_i)$ be a hypergraph consistent for $\mathbb{Q}_i$. Then, the choice of the set $\mathbb{Q}_{i+1}$ with the vertex v_i, as described above, will ensure that $|\mathbb{Q}_{i+1}| \geq \frac{|\mathbb{Q}_i|}{(6|V(H)|)^{|V(H)|+2}}$ and that $\mathcal{H}_{i+1}(\mathbb{Q}_{i+1})$ obtained from $\mathcal{H}_i(\mathbb{Q}_i)$ is consistent for $\mathbb{Q}_{i+1}$.

1) Representatives $\mathfrak{Rep}_i$ for $\mathbb{Q}$ and $\mathcal{H}_i(\mathbb{Q})$: In our analysis, we will be also using the concept of representatives to describe the scenario that a vertex from V has been contracted to some other vertices during the construction of $\mathcal{H}_i(\mathbb{Q})$ (in some moment, it has been deleted from $\mathcal{H}_j(\mathbb{Q})$, $1 \leq j < i$, and new hyperedges containing all neighbors of this vertex has been formed, in which case of these neighbors is used as a proxy). The canonical representative function plays an important role in our analysis and it is used explicitly in algorithms HTester and Random-HTraverse. (For the following definition, let us recall the construction of the hypergraph $\mathcal{H}_i(\mathbb{Q}_i)$ from Section VI-C. Let us also notice that the notion of canonical representatives is used solely in the analysis at the end of the process, and since it is not used for the construction of sets $\mathbb{Q}_1, \mathbb{Q}_2, \dots, \mathbb{Q}_{|V(H)|}$ and hypergraphs $\mathcal{H}_1(\mathbb{Q}_1), \mathcal{H}_2(\mathbb{Q}_2), \dots, \mathcal{H}_{|V(H)|}(\mathbb{Q}_{|V(H)|})$ and is used only to model their behavior, it does rely on the final order $v_1, \dots, v_{|V(H)|}$ of the vertices in H.)

Definition 34: **(Canonical representatives)** Let $\mathbb{Q}$ be a set of edge-disjoint colored copies of H in G. Let $v_1, \dots, v_{|V(H)|}$ be an arbitrary order of vertices of H such that for each i, $1 \leq i \leq |V(H)|$, the hypergraph $\mathcal{H}_i(\mathbb{Q})$ is consistent for $\mathbb{Q}$. A *canonical representative function* is a sequence of functions $\mathfrak{Rep}_1, \mathfrak{Rep}_2, \dots, \mathfrak{Rep}_{|V(H)|} : V \to V$ such that for every i, $1 \leq i \leq |V(H)|$:

- if u is an isolated vertex in $G[\mathbb{Q}]$, then $\mathfrak{Rep}_i(u) = u$ for every i;
- otherwise, if u is a vertex in $\mathcal{H}_i(\mathbb{Q})$ (i.e., $\chi(u) \notin \{\chi(v_j) : 1 \leq j < i\}$), then $\mathfrak{Rep}_i(u) = u$;
- otherwise, $\mathfrak{Rep}_i(u) = x$, where *(i)* $x \in \bigcup_{\mathfrak{e}: u \in \sigma(\mathfrak{e})} \mathfrak{e}$ and *(ii)* for any $x, y \in \bigcup_{\mathfrak{e}: u \in \sigma(\mathfrak{e})} \mathfrak{e}$, if $x \neq y$, $\chi(x) = \chi(v_{j_1})$, and $\chi(y) = \chi(v_{j_2})$, then $j_1 < j_2$.

We will denote any single $\mathfrak{Rep}_i$ as a *representative function*.

The notion of the canonical representative function $\mathfrak{Rep}_1, \mathfrak{Rep}_2, \dots, \mathfrak{Rep}_{|V(H)|} : V \to V$ describes the dependencies between the vertices from G in the construction of the sequence of the hypergraphs $\mathcal{H}_1(\mathbb{Q}), \mathcal{H}_2(\mathbb{Q}), \dots, \mathcal{H}_{|V(H)|}(\mathbb{Q})$. And so, $\mathfrak{Rep}_i(u) = u$ unless vertex u has been contracted during the construction of $\mathcal{H}_j(\mathbb{Q})$ for $j < i$. If u has been contracted during the construction of $\mathcal{H}_j(\mathbb{Q})$, then for some colored copy $\mathfrak{h}$ of H in $\mathbb{Q}$ containing u, we first added a new hyperedge consisting of vertices in $\mathcal{N}_j^{\mathfrak{h}}\langle u \rangle$, and then removed vertex u (with all incident edges from $\mathcal{H}_j(\mathbb{Q})$). In that case, we will define $\mathfrak{Rep}_j(u) = x$, [5]where x is the vertex in $\mathcal{N}_j^{\mathfrak{h}}\langle u \rangle$ that will be contracted first among all vertices in $\mathcal{N}_j^{\mathfrak{h}}\langle u \rangle$ (that is, if $x, y \in \mathcal{N}_j^{\mathfrak{h}}\langle u \rangle$ and $\chi(x) = \chi(v_{r_1})$ and $\chi(y) = \chi(v_{r_2})$, then $r_1 \leq r_2$). Furthermore, if in some future iteration $s > j$ vertex $x = \mathfrak{Rep}_j(u)$ is contracted, then we will not only set $\mathfrak{Rep}_s(x)$, but we will also update $\mathfrak{Rep}_s(u)$ to be the same as $\mathfrak{Rep}_s(x)$. In fact, we will maintain that for all $k > j$, if $\mathfrak{Rep}_j(u) = x$ then $\mathfrak{Rep}_k(u) = \mathfrak{Rep}_k(x)$.[6]

[5]Notice that this notion is well defined only since u is a safe vertex with respect to $\mathbb{Q}$ and $\mathcal{H}_j(\mathbb{Q})$, because in that case the neighbors of u in $\mathcal{H}_j(\mathbb{Q})$ do not depend on the choice of the copy $\mathfrak{h}$ of H in $\mathbb{Q}$ containing u we consider.

[6]Note that function $\mathfrak{Rep}_i$ defines a *forest* on V, where in each "tree" the root is a vertex u with $\mathfrak{Rep}_i(u) = u$, and the "leaves" are formed by vertices u with $\mathfrak{Rep}_i^{(-1)}(u) \neq u$ (that is, for which there is no v with $\mathfrak{Rep}_i(v) = u$).

Remark 35: Equivalently, one can define $\mathfrak{Rep}_1, \dots, \mathfrak{Rep}_{|V(H)|} : V \to V$ recursively as follows:

- if u is an isolated vertex in $G[\mathbb{Q}]$, then $\mathfrak{Rep}_i(u) = u$ for every i;
- otherwise:
  - ◇ $\mathfrak{Rep}_1(u) = u$ for every vertex $u \in V$;
  - ◇ for any i, $2 \le i \le |V(H)|$, for every $u \in V$:
    - ⋆ if u is a vertex in $\mathcal{H}_i(\mathbb{Q})$, then $\mathfrak{Rep}_i(u) = u$;
    - ⋆ otherwise,
      - ∗ if $\mathfrak{Rep}_{i-1}(u)$ has color different than $\chi(v_{i-1})$[7], then $\mathfrak{Rep}_i(u) = \mathfrak{Rep}_{i-1}(u)$;
      - ∗ else, $\mathfrak{Rep}_i(u)$ is equal to the neighbor of vertex $\mathfrak{Rep}_{i-1}(u)$ in $\mathcal{H}_{i-1}(\mathbb{Q})$ with the lowest color (that is, $\mathfrak{Rep}_i(u)$ is the neighbor x of $\mathfrak{Rep}_{i-1}(u)$ in $\mathcal{H}_{i-1}(\mathbb{Q})$ that minimizes j with $\chi(x) = \chi(v_j)$).

Let us explain the choice of vertex x in the last case of the definition of $\mathfrak{Rep}_i(u)$. First of all, the choice of $\mathfrak{Rep}_i(u)$ to be a neighbor of vertex $\mathfrak{Rep}_{i-1}(u)$ in $\mathcal{H}_{i-1}(\mathbb{Q})$ is to ensure that u will belong to the label of the newly created edge incident to that neighbor in $\mathcal{H}_i(\mathbb{Q})$. The choice of the neighbor with the "lowest color" is to ensure that that vertex will be the first to be contracted in the later procedure of shrinking $\mathcal{H}_j(\mathbb{Q})$, and thus, during that construction, the edge containing vertex u will be replaced by another edge. Therefore, our choosing x ensures that if $\chi(u) = \chi(v_r)$, then

- for every $i \le r$, $\mathfrak{Rep}_i(u) = u$, and
- for every $i > r$, $\mathfrak{Rep}_i(u)$ is a vertex in $\mathcal{H}_i(\mathbb{Q})$ and there is a hyperedge $\mathfrak{e}$ incident to vertex $\mathfrak{Rep}_i(u)$ such that $u \in \mathfrak{e}$.

VII. Completing the proof of Lemma 17, and of Theorems 14 and 15

We are now ready to complete the proof of Lemma 17, and with this of Theorems 14 and 15.

Let $G = (V, E)$ be a simple planar graph that is ε-far from H-free. By our analysis in the previous sections (see Lemma 33), we know that we can order the vertices of H $v_1, \dots, v_{|V(H)|}$ to define the hypergraphs $\mathcal{M}_1, \dots, \mathcal{M}_{|V(H)|}$, so that there are sets $\mathbb{Q}_1, \dots, \mathbb{Q}_{|V(H)|}$ of edge-disjoint colored copies of H in G with $\mathbb{Q}_{|V(H)|} \subseteq \mathbb{Q}_{|V(H)|-1} \subseteq \dots \subseteq \mathbb{Q}_1$ and $|\mathbb{Q}_{|V(H)|}| = \Omega_{\varepsilon,H}(|V|)$, such that for each i, $1 \le i \le |V(H)|$, the hypergraph $\mathcal{H}_i(\mathbb{Q}_i)$ is consistent for $\mathbb{Q}_i$.

Let us first apply Lemma 18 to the set $\mathbb{Q}_{|V(H)|}$ of edge-disjoint colored copies of H in G to obtain a subset $\mathbb{Q} \subseteq \mathbb{Q}_{|V(H)|}$ with $|\mathbb{Q}| = \Omega_{\varepsilon,H}(|V|)$, such that the graph $G[\mathbb{Q}]$ satisfies condition (a) of Lemma 17. Therefore, we only have to show that condition (b) of Lemma 17 holds too, that is, we have to show that if $G = (V, E)$ is a simple planar graph that is ε-far from H-free, then

> ⊗ Tester$(G[\mathbb{Q}], H, \mathfrak{d}, \mathfrak{t})$ finds a copy of H in $G[\mathbb{Q}]$ with probability $\Omega_{\varepsilon,H}(1)$.

$\mathbb{Q}$ is a set of edge-disjoint colored copies of H in G such that $|\mathbb{Q}| = \Omega_{\varepsilon,H}(|V|)$, and such that for each i, $1 \le i \le |V(H)|$, the hypergraph $\mathcal{H}_i(\mathbb{Q})$ is consistent for $\mathbb{Q}$. Let us take the canonical representative function $\mathfrak{Rep}_1, \mathfrak{Rep}_2, \dots, \mathfrak{Rep}_{|V(H)|} : V \to V$, cf. Definition 34.

We will prove ⊗ by showing the following two properties (proven below as Claims 36 and 37):

> 1. the probability that HTester $(\mathcal{H}_{|V(H)|}(\mathbb{Q}), \mathfrak{Rep}_{|V(H)|}, \mathcal{M}_{|V(H)|}, |V(H)|^2, 1)$ finds a copy of $\mathcal{M}_{|V(H)|}$ is $\Omega_{\varepsilon,H}(1)$, and
> 2. for every i, $1 \le i < |V(H)|$,
>   - if the probability that HTester $(\mathcal{H}_{i+1}(\mathbb{Q}), \mathfrak{Rep}_{i+1}, \mathcal{M}_{i+1}, \mathfrak{d}, \mathfrak{t})$ finds a copy of $\mathcal{M}_{i+1}$ is $\Omega_{\varepsilon,H}(1)$,
>   - then the probability that HTester $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathcal{M}_i, |V(H)| \cdot \mathfrak{d}, 2\mathfrak{t})$ finds a copy of $\mathcal{M}_i$ is $\Omega_{\varepsilon,H}(1)$.

Indeed, if Property 1 holds, then by iterating Property 2, we have that for some $\mathfrak{d}^*, \mathfrak{t}^* = \Omega_{\varepsilon,H}(1)$, the probability that HTester $(\mathcal{H}_1(\mathbb{Q}), \mathfrak{Rep}_1, \mathcal{M}_1, \mathfrak{d}^*, \mathfrak{t}^*)$ finds a copy of $\mathcal{M}_1$ is $\Omega_{\varepsilon,H}(1)$. Since $\mathfrak{Rep}_1$ is the identity function $\mathfrak{Rep}_1(u) = u$ for every $u \in V$, and since $\mathcal{H}_1(\mathbb{Q}) \equiv G[\mathbb{Q}]$, the behavior of Random-HTraverse $(\mathcal{H}_1(\mathbb{Q}), \mathfrak{Rep}_1, \mathfrak{d}^*, \mathfrak{t}^*)$ is identical to the behavior of Random-Traverse $(G[\mathbb{Q}], \mathfrak{d}^*, \mathfrak{t}^*)$, and further, since $\mathcal{M}_1 \equiv H$, the behavior of HTester $(\mathcal{H}_1(\mathbb{Q}), \mathfrak{Rep}_1, \mathcal{M}_1, \mathfrak{d}^*, \mathfrak{t}^*)$ is identical to the behavior of Tester $(G[\mathbb{Q}], H, \mathfrak{d}^*, \mathfrak{t}^*)$. Therefore, we obtain that the probability that Tester $(G[\mathbb{Q}], H, \mathfrak{d}^*, \mathfrak{t}^*)$ finds a copy of H is $\Omega_{\varepsilon,H}(1)$, what yields ⊗.

What remains is to prove that Properties 1 and 2 hold, what we do in the following two central claims, whose proofs are deferred to Section VII-A below.

Claim 36: The probability that algorithm HTester $(\mathcal{H}_{|V(H)|}(\mathbb{Q}), \mathfrak{Rep}_{|V(H)|}, \mathcal{M}_{|V(H)|}, |V(H)|^2, 1)$ finds a copy of $\mathcal{M}_{|V(H)|}$ is $\Omega_{\varepsilon,H}(1)$.

Claim 37: Let $1 \le i < |V(H)|$, $\mathfrak{d} = \mathfrak{d}(\varepsilon, H) \ge |V(H)|$, $\mathfrak{t} = \mathfrak{t}(\varepsilon, H)$, $\mathfrak{d}^* = |V(H)| \cdot \mathfrak{d}$, and $\mathfrak{t}^* = 2\mathfrak{t}$. If the probability that HTester $(\mathcal{H}_{i+1}(\mathbb{Q}), \mathfrak{Rep}_{i+1}, \mathcal{M}_{i+1}, \mathfrak{d}, \mathfrak{t})$ finds a copy of $\mathcal{M}_{i+1}$ is $\Omega_{\varepsilon,H}(1)$, then the probability that HTester $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathcal{M}_i, \mathfrak{d}^*, \mathfrak{t}^*)$ finds a copy of $\mathcal{M}_i$ is $\Omega_{\varepsilon,H}(1)$.

With Claims 36 and 37 at hand, we obtain that Properties 1 and 2 hold, and therefore we can conclude the proof of the proof of Lemma 17, and with this of Theorems 14 and 15. ■

A. Proofs of central Claims 36 and 37

In this section we give proofs of two our central results on which relies our proof of Lemma 17 (and with this of Theorems 14 and 15): Claims 36 and 37.

We begin with the proof of Claim 36.

Claim 36: The probability that algorithm HTester $(\mathcal{H}_{|V(H)|}(\mathbb{Q}), \mathfrak{Rep}_{|V(H)|}, \mathcal{M}_{|V(H)|}, |V(H)|^2, 1)$ finds a copy of $\mathcal{M}_{|V(H)|}$ is $\Omega_{\varepsilon,H}(1)$.

Proof. Our construction (see Section VI-B) ensures that $\mathcal{M}_{|V(H)|}$ has some number $\mathfrak{s}$ of hyperedges $\mathfrak{e}_1, \dots, \mathfrak{e}_{\mathfrak{s}}$, each $\mathfrak{e}_j$ consisting of a single vertex $v_{|V(H)|}$, and with the labels of edges $\mathfrak{e}_1, \dots, \mathfrak{e}_{\mathfrak{s}}$ defining a partition of $\{v_1, \dots, v_{|V(H)|-1}\}$ (that is, $\bigcup_{j=1}^{\mathfrak{s}} \sigma(\mathfrak{e}_j) = \{v_1, \dots, v_{|V(H)|-1}\}$ and $\sigma(\mathfrak{e}_{j_1}) \cap \sigma(\mathfrak{e}_{j_2}) = \emptyset$ for any $j_1 \ne j_2$).

[7] That is, $\mathfrak{Rep}_{i-1}(u)$ is not in $\mathcal{H}_{i-1}(\mathbb{Q})$.

Similarly, our construction (see Section VI-C) ensures that $\mathcal{H}_{|V(H)|}(\mathbb{Q})$ contains $\mathfrak{s} \cdot |\mathbb{Q}|$ hyperedges, each hyperedge $\mathfrak{e}$ in $\mathcal{H}_{|V(H)|}(\mathbb{Q})$ consisting of a single vertex of color $\chi(v_{|V(H)|})$. Furthermore, each such hyperedge $\mathfrak{e}$ corresponds (cf. Definition 23) to a copy of one of the hyperedges $\mathfrak{e}_1, \dots, \mathfrak{e}_\mathfrak{s}$ from $\mathcal{M}_{|V(H)|}$; let us denote by $\mathfrak{ind}(\mathfrak{e})$ the index of the copy $\mathfrak{e}_{\mathfrak{ind}(\mathfrak{e})}$ corresponding to $\mathfrak{e}$. Notice that $\sigma^*(\mathfrak{e}) = \sigma^*(\mathfrak{e}_{\mathfrak{ind}(\mathfrak{e})})$ and $|\{\mathfrak{e} \text{ in } \mathcal{H}_{|V(H)|}(\mathbb{Q}) : \mathfrak{ind}(\mathfrak{e}) = j\}| = |\mathbb{Q}|$ for any j, $1 \le j \le \mathfrak{s}$.

Let $\mathfrak{d}^* = |V(H)|^2$. In view of the comments and the construction above, by Definition 25, HTester $(\mathcal{H}_{|V(H)|}(\mathbb{Q}), \mathfrak{Rep}_{|V(H)|}, \mathcal{M}_{|V(H)|}, \mathfrak{d}^*, 1)$ finds a copy of $\mathcal{M}_{|V(H)|}$ if,

(1) in the call to Random-HTraverse $(\mathcal{H}_{|V(H)|}(\mathbb{Q}), \mathfrak{Rep}_{|V(H)|}, \mathfrak{d}^*, 1)$, it selects the starting vertex $u = \mathfrak{Rep}_{|V(H)|}(v)$ to be non-isolated in $\mathcal{H}_{|V(H)|}(\mathbb{Q})$, and

(2) vertex u chooses among its $\mathfrak{d}^*$ random incident edges all copies of $\mathfrak{e}_1, \dots, \mathfrak{e}_\mathfrak{s}$.

Our definition of $\mathfrak{Rep}$ ensures that $\mathfrak{Rep}_{|V(H)|}(x)$ is a non-isolated vertex in $\mathcal{H}_{|V(H)|}(\mathbb{Q})$ if and only if x is a non-isolated vertex in $G[\mathbb{Q}]$. Therefore we only have to show that $G[\mathbb{Q}]$ has $\Omega_{\varepsilon,H}(|V|)$ non-isolated vertices. Let $G^*[\mathbb{Q}]$ be the subgraph of $G[\mathbb{Q}]$ induced by non-isolated vertices. Since $G^*[\mathbb{Q}]$ consists of $|\mathbb{Q}|$ edge-disjoint copies of H, $G^*[\mathbb{Q}]$ has $|\mathbb{Q}| \cdot |E(H)|$ edges. Since $G^*[\mathbb{Q}]$ is a subgraph of a simple planar graph, $G^*[\mathbb{Q}]$ is a simple planar graph too, and thus must have at least $\frac{1}{3}|\mathbb{Q}| \cdot |E(H)|$ vertices. Therefore, since $|\mathbb{Q}| = \Omega_{\varepsilon,H}(|V|)$, we conclude that $G^*[\mathbb{Q}]$ has $\Omega_{\varepsilon,H}(|V|)$ vertices, or equivalently, that $G[\mathbb{Q}]$ has $\Omega_{\varepsilon,H}(|V|)$ non-isolated vertices. Therefore, with probability $\Omega_{\varepsilon,H}(1)$ Random-HTraverse $(\mathcal{H}_{|V(H)|}(\mathbb{Q}), \mathfrak{Rep}_{|V(H)|}, \mathfrak{d}^*, 1)$ selects a non-isolated as the starting vertex.

Next, let us condition on the fact that the starting vertex $u = \mathfrak{Rep}_{|V(H)|}(v)$ is non-isolated in $\mathcal{H}_{|V(H)|}(\mathbb{Q})$. Analogously to the classic coupon collector's problem, we can argue that if u selects at least $\mathfrak{s}^2$ (in fact, $\mathfrak{s}\ln(1+\mathfrak{s})$ would suffice too) times incident edges i.u.r., then with probability $\Omega_{\varepsilon,H}(1)$, the set $\mathcal{E}_{\ell,u}$ will contain $\mathfrak{s}$ hyperedges $\mathfrak{e}'_1, \dots, \mathfrak{e}'_\mathfrak{s}$ with $\mathfrak{ind}(\mathfrak{e}'_j) = j$ for every j, $1 \le j \le \mathfrak{s}$. In this case, the set $\mathcal{E}_{\ell,u}$ will contain a copy of $\mathcal{M}_{|V(H)|}$.

By our arguments above, this yields the claim. ■

We now move to the proof of Claim 37.

Claim 37: Let $1 \le i < |V(H)|$, $\mathfrak{d} = \mathfrak{d}(\varepsilon, H) \ge |V(H)|$, $\mathfrak{t} = \mathfrak{t}(\varepsilon, H)$, $\mathfrak{d}^* = |V(H)| \cdot \mathfrak{d}$, and $\mathfrak{t}^* = 2\mathfrak{t}$. If the probability that HTester $(\mathcal{H}_{i+1}(\mathbb{Q}), \mathfrak{Rep}_{i+1}, \mathcal{M}_{i+1}, \mathfrak{d}, \mathfrak{t})$ finds a copy of $\mathcal{M}_{i+1}$ is $\Omega_{\varepsilon,H}(1)$, then the probability that HTester $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathcal{M}_i, \mathfrak{d}^*, \mathfrak{t}^*)$ finds a copy of $\mathcal{M}_i$ is $\Omega_{\varepsilon,H}(1)$.

Proof. Let us refer to Definition 25 for the meaning of algorithm HTester $(\mathcal{H}_s(\mathbb{Q}), \mathfrak{Rep}_s, \mathcal{M}_s, \mathfrak{d}', \mathfrak{t}')$ (and thus also of Random-HTraverse $(\mathcal{H}_s(\mathbb{Q}), \mathfrak{Rep}_s, \mathfrak{d}', \mathfrak{t}')$) finding a colored copy of $\mathcal{M}_r$.

The proof relies on two basic properties that hold with probability $\Omega_{\varepsilon,H}(1)$:

- that a single step of Random-HTraverse $(\mathcal{H}_{i+1}(\mathbb{Q}), \mathfrak{Rep}_{i+1}, \mathfrak{d}, \mathfrak{t})$ can be simulated by 2 steps of Random-HTraverse $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathfrak{d}^*, \mathfrak{t}^*)$ with $\mathfrak{d}^* = |V(H)| \cdot \mathfrak{d}$ and $\mathfrak{t}^* = 2\mathfrak{t}$, and
- that if Random-HTraverse $(\mathcal{H}_{i+1}(\mathbb{Q}), \mathfrak{Rep}_{i+1}, \mathfrak{d}, \mathfrak{t})$ starts at a vertex u, then the same vertex u will be processed by Random-HTraverse $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathfrak{d}^*, \mathfrak{t}^*)$ in $L_0 \cup L_1$ (i.e., in one of the first two rounds).

Once these two claims hold, the proof of Claim 37 follows immediately.

We begin with showing that a single step of Random-HTraverse $(\mathcal{H}_{i+1}(\mathbb{Q}), \mathfrak{Rep}_{i+1}, \mathfrak{d}, \mathfrak{t})$ can be simulated by 2 steps of Random-HTraverse $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathfrak{d}^*, \mathfrak{t}^*)$.

We begin with two auxiliary definitions. For any pair of edges $\mathfrak{e}$ and $\mathfrak{e}'$, we say $\mathfrak{e}$ *and* $\mathfrak{e}'$ *are semi-equivalent* if their vertex sets are the same and their colored labels are the same. Let $\mathfrak{e}$ be an edge in $\mathcal{H}_{i+1}(\mathbb{Q})$ that is modeled by edges $\mathfrak{e}_1, \dots, e_\mathfrak{r}$ in $\mathcal{H}_i(\mathbb{Q})$ (cf. Definition 24). Then any $\mathfrak{r}$ edges $\mathfrak{e}'_1, \dots, e'_\mathfrak{r}$ in $\mathcal{H}_i(\mathbb{Q})$ are called *sub-equivalent to* $\mathfrak{e}$ if for every $1 \le j \le \mathfrak{r}$, edges $\mathfrak{e}_j$ and $\mathfrak{e}'_j$ are semi-equivalent.

The first definition relates to the scenario when HTester $(\mathcal{H}, \mathfrak{Rep}, \mathcal{M}_j, \mathfrak{d}, \mathfrak{t})$ finds a colored copy of $\mathcal{M}_j$ in $\mathcal{H}$ that contains edge $\mathfrak{e}$ in $\mathcal{H}$. In that case, we claim that the algorithm would have found a copy of $\mathcal{M}_j$ also if instead of using edge $\mathfrak{e}$, it used any edge semi-equivalent to $\mathfrak{e}$. The second definition is used to describe the scenario when HTester $(\mathcal{H}_{i+1}(\mathbb{Q}), \mathfrak{Rep}_{i+1}, \mathcal{M}_{i+1}, \mathfrak{d}, \mathfrak{t})$ finds a colored copy of $\mathcal{M}_{i+1}$ by finding edges $\mathcal{E}$ in $\mathcal{H}_{i+1}(\mathbb{Q})$ matching $\mathcal{M}_{i+1}$. In that case, to find a colored copy of $\mathcal{M}_i$, it is enough that HTester $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathcal{M}_i, \mathfrak{d}^*, \mathfrak{t}^*)$ finds only edges $\mathcal{E}'$ such that for every $\mathfrak{e} \in \mathcal{E}$, $\mathcal{E}'$ contains edges $\mathfrak{e}'_1, \dots, e'_\mathfrak{s}$ in $\mathcal{H}_i(\mathbb{Q})$ that are sub-equivalent to $\mathfrak{e}$.

Let us consider a step of creating set L_ℓ in Random-HTraverse $(\mathcal{H}_{i+1}(\mathbb{Q}), \mathfrak{Rep}_{i+1}, \mathfrak{d}, \mathfrak{t})$, and let u be a vertex in $L_{\ell-1}$ with incident edge $\mathfrak{e}$. Let $\mathfrak{e}$ belong to a copy $\mathfrak{h}_\mathfrak{e}$ of $\mathcal{M}_{i+1}$ in $\mathcal{H}_{i+1}(\mathbb{Q})$ and let $\widehat{\mathfrak{e}}$ be the corresponding edge in $\mathcal{M}_{i+1}$. By our construction, edge $\mathfrak{e}$ was either already present in $\mathcal{H}_i(\mathbb{Q})$, or is a result of a contraction in $\mathcal{H}_i(\mathbb{Q})$ of a vertex x with $\chi(x) = \chi(v_i)$. In the latter case, $\mathfrak{e}$ is equal to $\mathcal{N}_i^{\mathfrak{h}_\mathfrak{e}}\langle x \rangle$, the set of neighbors of x in $\mathfrak{h}_\mathfrak{e}$ (in $\mathcal{H}_i(\mathbb{Q}_i)$) other than x.

In Random-HTraverse $(\mathcal{H}_{i+1}(\mathbb{Q}), \mathfrak{Rep}_{i+1}, \mathfrak{d}, \mathfrak{t})$, when vertex u selects $\mathfrak{d}$ incident edges i.u.r., the probability that u chooses $\mathfrak{e}$ among its $\mathfrak{d}$ incident edges in Random-HTraverse $(\mathcal{H}_{i+1}(\mathbb{Q}), \mathfrak{Rep}_{i+1}, \mathfrak{d}, \mathfrak{t})$ is $\mathfrak{p}_{u,\mathfrak{e}} = 1 - (1 - 1/\deg_{\mathcal{H}_{i+1}(\mathbb{Q})}(u))^\mathfrak{d}$, where $\deg_{\mathcal{H}_{i+1}(\mathbb{Q})}(u)$ is the number of edges incident to vertex u in $\mathcal{H}_{i+1}(\mathbb{Q})$.

If edge $\mathfrak{e}$ was already present in $\mathcal{H}_i(\mathbb{Q})$, then the probability that u chooses $\mathfrak{e}$ among its $\mathfrak{d}$ incident edges in Random-HTraverse $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathfrak{d}^*, \mathfrak{t}^*)$ is equal to $1 - (1 - 1/\deg_{\mathcal{H}_i(\mathbb{Q})}(u))^{\mathfrak{d}^*}$. Next, we notice that for any vertex x in $\mathcal{H}_{i+1}(\mathbb{Q})$, $\deg_{\mathcal{H}_{i+1}(\mathbb{Q})}(u) \le \deg_{\mathcal{H}_i(\mathbb{Q})}(u) \le |V(H)| \deg_{\mathcal{H}_{i+1}(\mathbb{Q})}(u)$. (Indeed, for any colored copy $\mathfrak{h}$ of H in $\mathbb{Q}$ that contains vertex x, if we contract in $\mathfrak{h}$ a neighbor of x in $\mathcal{H}_i(\mathbb{Q})$, then we remove up to $|V(H)|$ edges from $\mathcal{H}_i(\mathbb{Q})$ and add exactly one new edge.) This implies that with our setting $\mathfrak{d}^* = |V(H)| \cdot \mathfrak{d}$, we have $1 - (1 - 1/\deg_{\mathcal{H}_i(\mathbb{Q})}(u))^{\mathfrak{d}^*} \ge$

$1-(1-1/(|V(H)|\cdot\deg_{\mathcal{H}_{i+1}(\mathbb{Q})}(u)))^{|V(H)|\cdot\mathfrak{d}} = \Omega_{\varepsilon,H}(1-(1-1/\deg_{\mathcal{H}_{i+1}(\mathbb{Q})}(u))^{\mathfrak{d}})$. [8] Therefore, we can conclude that:

Case 1: if edge $\mathfrak{e}$ is present in $\mathcal{H}_i(\mathbb{Q})$ and in Random-HTraverse $(\mathcal{H}_{i+1}(\mathbb{Q}), \mathfrak{Rep}_{i+1}, \mathfrak{d}, \mathfrak{t})$, vertex u selects $\mathfrak{e}$ among its $\mathfrak{d}$ incident edges with probability $\mathfrak{p}_{u,\mathfrak{e}}$, then in Random-HTraverse $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathfrak{d}^*, \mathfrak{t}^*)$, vertex u selects $\mathfrak{e}$ among its $\mathfrak{d}^*$ incident edges with probability $\Omega_{\varepsilon,H}(\mathfrak{p}_{u,\mathfrak{e}})$.

The case when edge $\mathfrak{e}$ is not present in $\mathcal{H}_i(\mathbb{Q})$ and has been obtained as a contraction of vertex x with $\chi(x) = \chi(v_i)$, with $\mathfrak{e} = \mathcal{N}_i^{\mathfrak{h}_\mathfrak{e}}\langle x\rangle$, is more complicated.

Since $\mathcal{H}_{i+1}(\mathbb{Q})$ is consistent for $\mathbb{Q}$, vertex x is safe with respect to $\mathbb{Q}$ and $\mathcal{H}_i(\mathbb{Q})$. Let x be incident to $\deg_{\mathcal{H}_i(\mathbb{Q})}(x)$ edges in $\mathcal{H}_i(\mathbb{Q}_i)$ and note that $\chi(x) = \chi(v_i)$. By Remark 27, we can group edges incident to x in $\mathcal{H}_i(\mathbb{Q}_i)$ into $\mathfrak{r}$ groups of *the same size* each (equal to $\deg_{\mathcal{H}_i(\mathbb{Q})}(x)/\mathfrak{r}$), each group corresponding to a copy of one of the $\mathfrak{r}$ edges incident to v_i in $\mathcal{M}_i$, any two edges from the same group being semi-equivalent.

After contracting vertex x, we will create $\mathfrak{s} = \deg_{\mathcal{H}_i(\mathbb{Q})}(x)/\mathfrak{r}$ new edges $\mathfrak{e}_1, \dots, \mathfrak{e}_\mathfrak{s}$ in $\mathcal{H}_{i+1}(\mathbb{Q})$, each new edge with the same vertex set $\mathcal{N}_i\langle x\rangle$ that correspond to the set of neighbors of x in $\mathcal{H}_i(\mathbb{Q}_i)$, and having the same colored label. Thus all new edges $\mathfrak{e}_1, \dots, \mathfrak{e}_\mathfrak{s}$ are semi-equivalent. Furthermore, any $\mathfrak{e}'_1, \dots, \mathfrak{e}'_\mathfrak{r}$ incident to x in $\mathcal{H}_i(\mathbb{Q}_i)$ that are from $\mathfrak{r}$ different groups are sub-equivalent to every edge in $\mathfrak{e}_1, \dots, \mathfrak{e}_\mathfrak{s}$.

We will compare the probability that after arriving at vertex u, Random-HTraverse $(\mathcal{H}_{i+1}(\mathbb{Q}), \mathfrak{Rep}_{i+1}, \mathfrak{d}, \mathfrak{t})$ visits any of the edges $\mathfrak{e}_1, \mathfrak{e}_2, \dots, \mathfrak{e}_\mathfrak{s}$, with the probability that after arriving at u, algorithm Random-HTraverse $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathfrak{d}^*, \mathfrak{t}^*)$ visits in $\mathcal{H}_i(\mathbb{Q})$ $\mathfrak{r}$ edges that are incident to x in $\mathcal{H}_i(\mathbb{Q}_i)$ and that are from $\mathfrak{r}$ different groups (and hence are sub-equivalent to every edge in $\mathfrak{e}_1, \dots, \mathfrak{e}_\mathfrak{s}$).

In Random-HTraverse, when vertex u selects $\mathfrak{d}$ incident edges i.u.r., the probability that it chooses at least one of the edges $\mathfrak{e}_1, \dots, \mathfrak{e}_\mathfrak{s}$ among its $\mathfrak{d}$ incident edges in Random-HTraverse $(\mathcal{H}_{i+1}(\mathbb{Q}), \mathfrak{Rep}_{i+1}, \mathfrak{d}, \mathfrak{t})$ is equal to $\mathfrak{p}_{i+1} = 1-(1-\mathfrak{s}/\deg_{\mathcal{H}_{i+1}(\mathbb{Q})}(u))^{\mathfrak{d}}$.

Let us compare it to the probability that in Random-HTraverse $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathfrak{d}^*, \mathfrak{t}^*)$, when vertex u selects $\mathfrak{d}$ incident edges i.u.r. then one of these edges is incident to vertex x, *and* when in Random-HTraverse $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathfrak{d}^*, \mathfrak{t}^*)$ vertex x selects $\mathfrak{d}$ incident edges i.u.r. then at least one edge from each of the $\mathfrak{r}$ groups of edges incident to x in $\mathcal{H}_i(\mathbb{Q})$ is chosen[9]. The first probability, that one of the incident edges selected by u is incident to x, is equal to $\mathfrak{p}_i \ge 1-(1-\mathfrak{s}/\deg_{\mathcal{H}_i(\mathbb{Q})}(u))^{\mathfrak{d}^*}$, since the number of edges containing both u and x in $\mathcal{H}_i(\mathbb{Q})$ is at least $\mathfrak{s}$. To estimate the second probability, similarly as we were already arguing in the proof of Claim 36 and analogously to the classic coupon collector's problem, if x selects at least $\mathfrak{r}^2$ (in fact, $\mathfrak{r}\ln(1+\mathfrak{r})$ would suffice too) times incident edges i.u.r. (and we have $\mathfrak{d}^* \ge |V(H)|^2$), then with probability $\Omega_{\varepsilon,H}(1)$, the corresponding set $\mathcal{E}_{\cdot,x}$ will contain at least one edge from each of the $\mathfrak{r}$ groups of edges incident to x in $\mathcal{H}_i(\mathbb{Q})$. Therefore, in summary, with probability $\Omega_{\varepsilon,H}(\mathfrak{p}_i)$, if Random-HTraverse $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathfrak{d}^*, \mathfrak{t}^*)$ visits vertex u, then the algorithm will visit (until at most two rounds later) edges $\mathfrak{e}'_1, \dots, \mathfrak{e}'_\mathfrak{r}$ that are sub-equivalent to edges $\mathfrak{e}_1, \dots, \mathfrak{e}_\mathfrak{s}$.

Now we only have to match the probabilities of these events in Random-HTraverse $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathfrak{d}^*, \mathfrak{t}^*)$ and in Random-HTraverse $(\mathcal{H}_{i+1}(\mathbb{Q}), \mathfrak{Rep}_{i+1}, \mathfrak{d}, \mathfrak{t})$. Since, as we were arguing above, $\deg_{\mathcal{H}_{i+1}(\mathbb{Q})}(u) \le \deg_{\mathcal{H}_i(\mathbb{Q})}(u) \le |V(H)|\deg_{\mathcal{H}_{i+1}(\mathbb{Q})}(u)$, we note that with our setting $\mathfrak{d}^* = |V(H)|\cdot\mathfrak{d}$, we have $\mathfrak{p}_{i+1} = 1-(1-\mathfrak{s}/\deg_{\mathcal{H}_{i+1}(\mathbb{Q})}(u))^{\mathfrak{d}} = \Omega_{\varepsilon,H}(\mathfrak{p}_i)$, using the same arguments as before. This gives the following:

Case 2: if edge $\mathfrak{e}$ is not in $\mathcal{H}_i(\mathbb{Q})$, when Random-HTraverse $(\mathcal{H}_{i+1}(\mathbb{Q}), \mathfrak{Rep}_{i+1}, \mathfrak{d}, \mathfrak{t})$ arrives at vertex u, if $\mathfrak{p}_{i+1}$ is the probability that u selects an edge semi-equivalent to $\mathfrak{e}$ among its $\mathfrak{d}$ incident edges, then when Random-HTraverse $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathfrak{d}^*, \mathfrak{t}^*)$ arrives at u (with $u \in L_\ell$), then with probability $\Omega_{\varepsilon,H}(\mathfrak{p}_{i+1})$ the set $\bigcup_{j=1}^{\ell+2}\mathcal{E}_j$ of selected edges until at most two rounds later contains edges $\mathfrak{e}'_1, \dots, \mathfrak{e}'_\mathfrak{r}$ that are sub-equivalent to $\mathfrak{e}$.

Therefore, in summary, our analysis of Case 1 and Case 2 above implies our claim that a single step of algorithm Random-HTraverse $(\mathcal{H}_{i+1}(\mathbb{Q}), \mathfrak{Rep}_{i+1}, \mathfrak{d}, \mathfrak{t})$ can be simulated by 2 steps of algorithm Random-HTraverse $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathfrak{d}^*, \mathfrak{t}^*)$, with the success probability loss of $O_{\varepsilon,H}(1)$. That is, if one arrives at vertex u in step k of Random-HTraverse $(\mathcal{H}_{i+1}(\mathbb{Q}), \mathfrak{Rep}_{i+1}, \mathfrak{d}, \mathfrak{t})$ and the probability that one selects an edge semi-equivalent to $\mathfrak{e}$ is $\mathfrak{p}_{i+1}$, then if one arrives at vertex u in step ℓ of Random-HTraverse $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathfrak{d}^*, \mathfrak{t}^*)$, then with probability $\Omega_{\varepsilon,H}(\mathfrak{p}_{i+1})$, either $\mathcal{E}_{\ell+1}$ contains an edge semi-equivalent to $\mathfrak{e}$, or $\bigcup_{j=1}^{\ell+2}\mathcal{E}_j$ contains edges $\mathfrak{e}'_1, \dots, \mathfrak{e}'_\mathfrak{r}$ that are sub-equivalent to $\mathfrak{e}$.

Choosing starting vertex: Let us recall that the probability to choose $u \in V(\mathcal{H}_{i+1}(\mathbb{Q}))$ as a starting vertex $\mathfrak{Rep}_{i+1}(v)$ in Random-HTraverse $(\mathcal{H}_{i+1}(\mathbb{Q}), \mathfrak{Rep}_{i+1}, \mathfrak{d}, \mathfrak{t})$ is $\mathfrak{p}_{i+1} = \frac{|\mathfrak{Rep}_{i+1}^{(-1)}(u)|}{|V|}$. Since we may contract many vertices into

[8]To see this, think about the following experiment. Choosing $\mathfrak{e}$ in $\mathcal{H}_{i+1}(\mathbb{Q})$ is like choosing one out of $\deg_{\mathcal{H}_{i+1}(\mathbb{Q})}(u)$ incident edges, and repeating it $\mathfrak{d}$ times; choosing $\mathfrak{e}$ in $\mathcal{H}_i(\mathbb{Q})$ is like choosing one out of up to $|V(H)|\cdot\deg_{\mathcal{H}_{i+1}(\mathbb{Q})}(u)$ incident edges, and repeating it $\mathfrak{d}^*$ times. Now, to choose $\mathfrak{e}$ in $\mathcal{H}_i(\mathbb{Q})$ we can also split all edges incident to u in $\mathcal{H}_i(\mathbb{Q})$ into $\deg_{\mathcal{H}_{i+1}(\mathbb{Q})}(u)$ groups, each group of size approximately $\deg_{\mathcal{H}_i(\mathbb{Q})}(u)/\deg_{\mathcal{H}_{i+1}(\mathbb{Q})}(u)$. Then, the probability that we will choose an edge from the same group as $\mathfrak{e}$ is $\mathfrak{p}_{u,\mathfrak{e}}$ (approximately, because of rounding) the same as the probability that we will choose edge $\mathfrak{e}$ in $\mathcal{H}_{i+1}(\mathbb{Q})$. Therefore, with probability at most $1/|V(H)|$, we would then choose edge $\mathfrak{e}$ in $\mathcal{H}_i(\mathbb{Q})$. If we repeat this $|V(H)|$ time, we will get probability $\Omega_{\varepsilon,H}(\mathfrak{p}_{u,\mathfrak{e}})$. (Notice that we could also be happy with the probability $\mathfrak{p}_{u,\mathfrak{e}}/|V(H)|$, since this is $\Omega_{\varepsilon,H}(\mathfrak{p}_{u,\mathfrak{e}})$.)

[9]Let us notice that we do not assume that x will be processed in the *next round* in Random-HTraverse $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathfrak{d}^*, \mathfrak{t}^*)$, after vertex u is processed. This is because it is possible that vertex x has been processed before vertex u, for example, as the very first vertex in the call to Random-HTraverse $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathfrak{d}^*, \mathfrak{t}^*)$. Our arguments imply that both u and x will be processed (in the way we want them to be processed) *not later* than in the next round.

u during our construction, the probability of choosing u as a starting vertex in Random-HTraverse $(\mathcal{H}_{i+1}(\mathbb{Q}), \mathfrak{Rep}_{i+1}, \mathfrak{d}, \mathfrak{t})$ can be significantly larger than the probability of choosing u in Random-HTraverse $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathfrak{d}^*, \mathfrak{t}^*)$, which is $\mathfrak{p}_i = \frac{|\mathfrak{Rep}_i^{(-1)}(u)|}{|V|}$. However, our definition of $\mathfrak{Rep}_{i+1}$ ensures that

$$|\mathfrak{Rep}_{i+1}^{(-1)}(u)| = |\mathfrak{Rep}_i^{(-1)}(u)| + \sum_{\substack{x \text{ adjacent to } u \text{ in } \mathcal{H}_i(\mathbb{Q}): \\ \chi(x)=\chi(v_i)}} |\mathfrak{Rep}_i^{(-1)}(x)| \enspace .$$

Let us notice that if a vertex x of color $\chi(v_i)$ that is adjacent to u in $\mathcal{H}_i(\mathbb{Q})$ is selected as the starting vertex in Random-HTraverse $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathfrak{d}^*, \mathfrak{t}^*)$, which happens with probability $\frac{|\mathfrak{Rep}_i^{(-1)}(x)|}{|V|}$, then since (cf. Lemma 33) x is a safe vertex with respect to $\mathbb{Q}$ and $\mathcal{H}_i(\mathbb{Q})$, each copy of $\mathcal{M}_i$ in $\mathcal{H}_i(\mathbb{Q})$ containing vertex x has at least one edge containing also vertex u. Therefore, in Random-HTraverse $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathfrak{d}^*, \mathfrak{t}^*)$, we will not only have $x \in L_0$, but also if $\mathfrak{d} = \Omega_{\varepsilon,H}(1)$ is sufficiently large ($\mathfrak{d} > |V(H)|$ will suffice), then with probability at least $\frac{1}{2}$ we will have $u \in L_1$. Summing up over all starting vertices (including u), we obtain that u is in $L_0 \cup L_1$ with probability at least $\frac{1}{2}\mathfrak{p}_{i+1}$.

Now we are ready to complete the analysis and prove Claim 37. Let us consider the random process Random-HTraverse $(\mathcal{H}_{i+1}(\mathbb{Q}), \mathfrak{Rep}_{i+1}, \mathfrak{d}, \mathfrak{t})$ selecting vertices and edges to define L_j and $\mathcal{E}_{j+1}$ for $0 \le j \le \mathfrak{t}$. Similarly, let us consider the random process of Random-HTraverse $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathfrak{d}^*, 2\mathfrak{t})$ selecting vertices and edges to define L'_j and $\mathcal{E}'_{j+1}$ for $0 \le j \le 2\mathfrak{t}$. Notice that $|\bigcup_{j=0}^{\mathfrak{t}} L_j| = \Omega_{\varepsilon,H}(1)$, $|\bigcup_{j=1}^{\mathfrak{t}} \mathcal{E}_j| = \Omega_{\varepsilon,H}(1)$, $|\bigcup_{j=0}^{2\mathfrak{t}} L'_j| = \Omega_{\varepsilon,H}(1)$, $|\bigcup_{j=1}^{2\mathfrak{t}} \mathcal{E}'_j| = \Omega_{\varepsilon,H}(1)$. Suppose that HTester $(\mathcal{H}_{i+1}(\mathbb{Q}), \mathfrak{Rep}_{i+1}, \mathcal{M}_{i+1}, \mathfrak{d}, \mathfrak{t})$ starts at a vertex $u = \mathfrak{Rep}_{i+1}(v)$ and finds a copy of $\mathcal{M}_{i+1}$ consisting of edges $\mathfrak{e}_1, \dots, \mathfrak{e}_k$ in $\mathcal{H}_{i+1}(\mathbb{Q})$, where $k = |V(H)| - i$. Then, our analysis above gives that with at most a constant-factor probability loss, HTester $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathcal{M}_i, \mathfrak{d}^*, 2\mathfrak{t})$ will have u in $L'_0 \cup L'_1$, and then, for every edge $\mathfrak{e}_j$, $1 \le j \le k$, will either have $\mathfrak{e}_j \in \bigcup_{t=1}^{2\mathfrak{t}} \mathcal{E}'_t$ or $\mathfrak{e}'_{j_1}, \dots, \mathfrak{e}'_{j_\mathfrak{r}} \in \bigcup_{t=1}^{2\mathfrak{t}} \mathcal{E}'_t$, where $\mathfrak{e}'_{j_1}, \dots, \mathfrak{e}'_{j_\mathfrak{r}}$ are sub-equivalent to edges $\mathfrak{e}_j$ (this defines a proper coupling, properly taking care of multiple edges equivalent to $\mathfrak{e}_j$). Now, since every edge $\mathfrak{e}_j$

(i) either corresponds to an edge in both $\mathcal{M}_{i+1}$ and $\mathcal{M}_i$, or

(ii) corresponds to an edge $\widehat{\mathfrak{e}}$ in $\mathcal{M}_{i+1}$ that is modeled by $\widehat{\mathfrak{e}'_{j_1}}, \dots, \widehat{\mathfrak{e}'_{j_\mathfrak{r}}}$ in $\mathcal{M}_i$, and edges $\mathfrak{e}'_{j_1}, \dots, \mathfrak{e}'_{j_\mathfrak{r}}$ correspond to the edges $\widehat{\mathfrak{e}'_{j_1}}, \dots, \widehat{\mathfrak{e}'_{j_\mathfrak{r}}}$,

we can argue that in that case, HTester $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathcal{M}_i, \mathfrak{d}^*, 2\mathfrak{t})$ will find a copy of $\mathcal{M}_i$ (cf. Definition 25).

Therefore, with only a constant-factor probability loss, if HTester $(\mathcal{H}_{i+1}(\mathbb{Q}), \mathfrak{Rep}_{i+1}, \mathcal{M}_{i+1}, \mathfrak{d}, \mathfrak{t})$ finds a copy of $\mathcal{M}_{i+1}$ then HTester $(\mathcal{H}_i(\mathbb{Q}), \mathfrak{Rep}_i, \mathcal{M}_i, |V(H)|\mathfrak{d}, 2\mathfrak{t})$ finds a copy of $\mathcal{M}_i$. ■

VIII. Extension to families of arbitrary (not necessarily connected) finite graphs

Our result in Theorem 14 can be easily extended to allow the *forbidden finite graphs H to be arbitrary*, that is, not necessarily connected. Furthermore, the analysis extends in a straightforward way to the case when one wants to *test if for a given arbitrary finite family $\mathcal{H}$ of finite graphs, the input planar graph G is $\mathcal{H}$-free*, that is, contain no copy of any graph from $\mathcal{H}$.

Disconnected H: Notice that when H is not connected, Tester $(G, H, \mathfrak{d}, \mathfrak{t})$ may not be able to find a copy of H in G since it explores only a small connected neighborhood of the randomly sampled starting vertex v. However, one can easily extend the tester to be run separately on each connected component of H to do the job.

Let us assume that H consists of connected components $\mathfrak{h}_1, \mathfrak{h}_2, \dots, \mathfrak{h}_r$. As in Section IV-A1, we color the vertices of H arbitrarily, using $|V(H)|$ distinct colors $\{1, 2, \dots, |V(H)|\}$, one color for each vertex. Our analysis in Section IV starts with (an existential) Lemma 20 that if G is ε-far from H-free, then one can color vertices of G with $|V(H)|$ colors χ such that G has a set of $\Omega_{\varepsilon,H}(|V|)$ edge-disjoint colored copies of H. It is easy to see that Lemma 20 holds also for disconnected H. And so, in particular, for every connected component $\mathfrak{h}_i$ of H, there are $\Omega_{\varepsilon,H}(|V|)$ edge-disjoint colored copies of $\mathfrak{h}_i$ with colors of the vertices consistent with the coloring χ of G. Furthermore, since all connected components $\mathfrak{h}_1, \mathfrak{h}_2, \dots, \mathfrak{h}_r$ use distinct colors in H, these copies will be edge-disjoint between the copies of $\mathfrak{h}_1, \mathfrak{h}_2, \dots, \mathfrak{h}_r$. Then, for every connected component $\mathfrak{h}_i$ of H, we run Tester $(G, \mathfrak{h}_i, \mathfrak{d}_i, \mathfrak{t}_i)$, and the identical analysis as in Sections IV – VII concludes that Theorem 15 holds in the following way: there are positive functions $\mathfrak{d}_i = \mathfrak{d}_i(\varepsilon, \mathfrak{h}_i) = O_{\varepsilon,H}(1)$ and $\mathfrak{t}_i = \mathfrak{t}(\varepsilon, \mathfrak{h}_i) = O_{\varepsilon,H}(1)$, such that for any planar graph G that is ε-far from H-free, Tester$(G, \mathfrak{h}_i, \mathfrak{d}_i, \mathfrak{t}_i)$ finds a colored copy of $\mathfrak{h}_i$ with probability $\Omega_{\varepsilon,H}(1)$. Since the colored copies of connected components $\mathfrak{h}_1, \mathfrak{h}_2, \dots, \mathfrak{h}_r$ are pairwise disjoint in G, this implies that if we run Tester $(G, \mathfrak{h}_i, \mathfrak{d}_i, \mathfrak{t}_i)$ for $1 \le i \le r$, with appropriate $\mathfrak{d}_i = O_{\varepsilon,H}(1)$ and $\mathfrak{t}_i = O_{\varepsilon,H}(1)$, then for any planar graph G that is ε-far from H-free, we find a colored copy of H with probability $\Omega_{\varepsilon,H}(1)$. Therefore, if we repeat this process $O_{\varepsilon,H}(1)$ many times, we can amplify the error probability and obtain that for any planar graph G that is ε-far from H-free, we find a colored copy of H with probability at least $\frac{2}{3}$.

Forbidden family: Next, we extend our study to test if a given planar graph contains no copy of any forbidden graph from a given finite family of finite graphs. Let $\mathcal{H}$ be an arbitrary finite family of finite graphs (for a given $\varepsilon > 0$, we allow the size to be $O_\varepsilon(1)$). We say a simple graph G is *$\mathcal{H}$-free* if it is H-free for every $H \in \mathcal{H}$; G is *ε-far from $\mathcal{H}$-free* if one has to delete more than $\varepsilon|V|$ edges from G to obtain an $\mathcal{H}$-free graph. This definition implies that since $\mathcal{H}$ is finite, if G is ε-far from $\mathcal{H}$-free, then there is $H \in \mathcal{H}$ such that G is $\varepsilon/|\mathcal{H}|$-far from H-free.

Let us suppose that $\mathcal{H}$ is an arbitrary finite family of finite graphs. (Note that since $\mathcal{H}$ is a finite family of finite graphs, $|\mathcal{H}| = O_\varepsilon(1)$.) Then our analysis above can be easily extended to test with a constant number of queries if a planar graph is $\mathcal{H}$-free. Indeed, let us run a constant query-time $\varepsilon/|\mathcal{H}|$-tester for every $H \in \mathcal{H}$, and reject if any of the tests rejects. Notice that if G is $\mathcal{H}$-free then this tester will accept, and if G is ε-far from $\mathcal{H}$-free then since there is $H \in \mathcal{H}$ such that G is $\varepsilon/|\mathcal{H}|$-far from H, the tester will reject G with probability at least $\frac{2}{3}$.

The discussion above can be summarized in the following theorem.

Theorem 38: Let $\mathcal{H}$ be an arbitrary collection of (not necessarily connected) finite graphs. Then there is a one-sided error property tester that for any simple planar graph G performs a constant number of queries to the random neighbor oracle and accepts if G is $\mathcal{H}$-free, and with probability at least $\frac{2}{3}$ rejects if G is ε-far from $\mathcal{H}$-free.

Theorem 38 holds also if $\mathcal{H}$ varies with different ε. That is, if for a given $\varepsilon > 0$, the goal is to test if G is $\mathcal{H}$-free or is ε-far from $\mathcal{H}$-free, for a finite family of graphs $\mathcal{H}$ that may depend on ε.

IX. Extending the analysis to minor-free graphs

While throughout the paper we focused on testing H-freeness of *planar graphs*, our techniques can easily be extended to any *class of minor-free graphs*. Recall that a graph L is called a *minor* of a graph G if L can be obtained from G via a sequence of vertex and edge deletions, and edge contractions. For any graph L, a graph G is called *L-minor-free* if L is not a minor of G. (For example, by Kuratowski's Theorem, a graph is planar if and only if it is $K_{3,3}$-minor-free and K_5-minor-free.)

Let us fix a graph L and consider the input graph G to be an L-minor-free graph. We now argue now that entire analysis presented in the previous sections easily extends to testing H-freeness of G. The key observation is that our analysis in Sections III–VII relies only on the following two properties of planar graphs:

(i) every minor of a planar graph is planar,
(ii) the number of edges in a planar graph is $O(n)$, where n is the number of vertices

It is known that these two properties hold for any class of L-minor-free graphs (that is, the first property would be that every minor of an L-minor-free graph is L-minor-free). Therefore, we can proceed with nearly identical analysis for L-minor-free graphs and arrive at the following version of Theorem 14.

Theorem 39: Let L be a fixed graph. There are positive functions f, g, and h such that for any L-minor-free-graph G:

- if G is H-free, then Random-Exploration(G, H, ε) accepts G, and
- if G is ε-far from H-free, then Random-Exploration(G, H, ε) rejects G with probability at least 0.99.

Furthermore, in the same way as in Section VIII, we can extend Theorem 38 to obtain the following.

Theorem 40: Let L be a fixed graph. Let $\mathcal{H}$ be an arbitrary collection of (not necessarily connected) finite graphs. Then there is a one-sided error property tester that for any L-minor-free-graph G performs a constant number of queries to the random neighbor oracle and accepts if G is $\mathcal{H}$-free, and with probability at least $\frac{2}{3}$ rejects if G is ε-far from $\mathcal{H}$-free.

Remark 41: It should be noted that while our main focus is on the random neighbor oracle model, it is straightforward to extend our testers (and their analysis) for $\mathcal{H}$-freeness to the other three oracle access model presented in Section I-B. Indeed, since each of these models can trivially simulate the random neighbor oracle model without any loss in the query complexity, Theorem 40 (and also Theorems 14 and 38) holds also for all these oracle access models.

However, our main result, the characterization of testable properties in planar graphs, as well as our reduction in Theorem 12, cannot be extended to the other models (see Section I-C4).

X. Conclusions

The fundamental problem in the area of property testing is to understand the complexity of testing graph properties in all natural models. One of the central questions here is to provide characterizations of testable graph properties in these models, that is, to determine which graph properties can be tested with constant query complexity. While we have characterizations of graph properties testable in the dense graph model, and some understanding of testable graph properties in the bounded-degree graph model, finding such a characterization in a very natural case of general graphs, without any bounds for their maximum degrees, remains a challenging and elusive open problem. The main result of this paper, Theorem 5, resolves an important natural special case of this open problem, which concerns property testers for planar graphs and for minor-closed graphs with one-sided error in the random neighbor oracle model.

Our main technical, algorithmic contribution significantly extend the approach from [11] to prove that H-freeness is testable with a constant number of queries for general planar graphs. Our result was proven via a new type of analysis of random exploration of planar graphs and their combination of the study of hypergraph representations of contractions in planar graphs. Our analysis easily carries over to classes of graphs defined by general fixed forbidden minors.

Our work is a continuation of our efforts to understand the complexity of testing basic graph properties in graphs with no bounds for the degrees. Indeed, while major efforts in the property testing community have been put to study dense graphs and bounded degree graphs (cf. [16, Chapter 8-9]), we have seen only limited advances in the study of general graphs, in particular, sparse graphs but without any bounds for the maximum degrees. We believe that this model is one of the most natural models, and it is also most relevant to computer science applications. Similarly as it has been done in [16, Chapter 10.5.3], we would advocate further study of this model because of its importance, its applications, and the variety (and beauty) of techniques used to advance this topic.

REFERENCES

[1] N. Alon, E. Fischer, M. Krivelevich, and M. Szegedy. Efficient testing of large graphs. *Combinatorica*, 20:451–476, 2000.

[2] N. Alon, E. Fischer, I. Newman, and A. Shapira. A combinatorial characterization of the testable graph properties: it's all about regularity. *SIAM Journal on Computing*, 39:143–167, 2009.

[3] N. Alon, T. Kaufman, M. Krivelevich, and D. Ron. Testing triangle-freeness in general graphs. *SIAM Journal on Discrete Mathematics*, 22(2):786–819, 2008.

[4] N. Alon and M. Krivelevich. Testing k-colorability. *SIAM Journal on Discrete Mathematics*, 15(2):211–227, 2002.

[5] N. Alon and A. Shapira. Every monotone graph property is testable. *SIAM Journal on Computing*, 38(2):505–522, 2008.

[6] N. Alon and A. Shapira. A characterization of the (natural) graph properties testable with one-sided error. *SIAM Journal on Computing*, 37(6):1703–1727, 2008.

[7] N. Alon and A. Shapira. A separation theorem in property testing. *Combinatorica*, 28(3): 261–281, 2008.

[8] I. Benjamini, O. Schramm, and A. Shapira. Every minor-closed property of sparse graphs is testable. *Advances in Mathematics*, 223:2200–2218, 2010.

[9] A. Czumaj, H. Fichtenberger, P. Peng, and C. Sohler Testable properties in general graphs and random order streaming. *CoRR abs/1905.01644*, 2019.

[10] A. Czumaj, O. Goldreich, D. Ron, C. Seshadhri, A. Shapira, and C. Sohler. Finding cycles and trees in sublinear time. *Random Structures and Algorithms*, 45(2): 139–184, 2014.

[11] A. Czumaj, M. Monemizadeh, K. Onak, and C. Sohler. Planar graphs: Random walks and bipartiteness testing. In *Proceedings of the 52th IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 423–432, 2011.

[12] A. Czumaj, A. Shapira, and C. Sohler. Testing hereditary properties of nonexpanding bounded-degree graphs. *SIAM Journal on Computing*, 38(6): 2499–2510, April 2009.

[13] A. Czumaj and C. Sohler. A characterization of graph properties testable for general planar graphs with one-sided error (It is all about forbidden subgraphs). A full version of the current paper, *CoRR abs/1909.10647*.

[14] T. Eden, A. Levi, D. Ron, and C. Seshadhri. Approximately counting triangles in sublinear time. *SIAM Journal on Computing*, 46(5): 1603–1646, 2017.

[15] T. Eden, D. Ron, and C. Seshadhri. On approximating the number of k-cliques in sublinear time. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 722–734, 2018.

[16] O. Goldreich. *Introduction to Property Testing*. Cambridge University Press 2017.

[17] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4): 653–750, July 1998.

[18] O. Goldreich and D. Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2): 302–343, 2002.

[19] O. Goldreich and D. Ron. A sublinear bipartiteness tester for bounded degree graphs. *Combinatorica*, 19(3):335–373, 1999.

[20] M. Gonen, D. Ron, and Y. Shavitt. Counting stars and other small subgraphs in sublinear-time. *SIAM Journal on Discrete Mathematics*, 25(3): 1365–1411, 2011.

[21] H. Ito. Every property is testable on a natural class of scale-free multigraphs. In *Proceedings of the 24th Annual European Symposium on Algorithms (ESA)*, pp. 51:1–12, 2016.

[22] T. Kaufman, M. Krivelevich, and D. Ron. Tight bounds for testing bipartiteness in general graphs. *SIAM Journal on Computing*, 33(6): 1441–1483, September 2004.

[23] S. Marko and D. Ron. Approximating the distance to properties in bounded-degree and general sparse graphs. *ACM Transactions on Algorithms*, 5(2), Article No. 22, March 2009.

[24] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298(5594): 824–827 2002.

[25] I. Newman and C. Sohler. Every property of hyperfinite graphs is testable. *SIAM Journal on Computing*, 42(3): 1095–1112, 2013.

[26] M. Parnas and D. Ron. Testing the diameter of graphs. *Random Structures & Algorithms*, 20(2):165–183, 2002.

[27] R. Rubinfeld and M. Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.